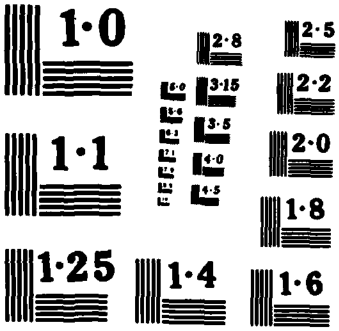
AD-A158 285 DATA BASE MANAGEMENT: PROCEEDINGS OF A CONFERENCE 1/4
NOVEMBER 1-2 1984 HELD AT MONTEREY CALIFORNIA(U)
STANFORD UNIV CA DEPT OF STATISTICS H SOLOMON ET AL.
UNCLASSIFIED 31 JUL 85 N00014-76-C-0475 F/G 9/2 NL


1·0
2·8
2·5
3·15
2·2
3·5
1·1
2·0
4·0
4·5
1·8
1·25
1·4
1·6

[illegible]

N00014-7[illegible]-0475 ([illegible]-161)

FOR THE OFFICE OF NAVAL RESEARCH

FOREWORD

Stanford University and the Navy Center for International Science and Technology (NCIST) and Office of Naval Research unit, located at the U.S. Naval Postgraduate School (NPGS) in Monterey jointly sponsored a Database Management Conference on November 1-2, 1984. The meeting was held at the NPGS. Eleven speakers were invited to give talks over the two days to audiences numbering 25 to 35 workers in the field. These speakers covered a wide spectrum of specialties in database management. They are listed together with their topics on the Conference Schedule that follows this introduction. The papers are printed in the order given at the conference.

Eight speakers provided us with written versions of their talks, which appear in the report. Other commitments precluded Drs. Lenat, Resnikoff, and Schuler from submitting written accounts of their talks. Members of the audience were workers actively involved in database management who provided lively comments and questions. We thank all the participants for their efforts.

The co-directors of the Conference were Professor Herbert Solomon of Stanford University, and Dr. Elliot H. Weinberg, Director of NCIST. Funds for the meeting were provided by the Office of Naval Research.

This compilation should be assigned one AD number per Ms. Rothwell, ONR/Code 411SP

DATA BASE MANAGEMENT:

PROCEEDINGS OF A CONFERENCE, NOVEMBER 1-2, 1984

Prepared Under Contract
N00014-76-C-0475 (NR-042-267)
For the Office of Naval Research

July 31, 1985

Herbert Solomon, Project Director

DEPARTMENT OF STATISTICS
STANFORD UNIVERSITY
STANFORD, CALIFORNIA

Data Base Management Conference

SCHEDULE

Thursday, November 1

8:30	Registration and Coffee	
9:00	Introduction and Welcome	
9:15	Gio Wiederhold Stanford University Stanford, California	Database Organization for Statistical Data Analysis.
10:15	Refreshments	
10:30	Robert Blum Stanford University Stanford, California	Automating the Study of Causal Relationships ... Large Time-Oriented Clinical Databases: The RX Project
11:30	James Dewar and James Gillogly Applied Sciences and Engineering Division, Rand Corporation Santa Monica, California	CODA: A Concept Organization and Development Aide for the Research Environment
12:30	Lunch	
2:00	John McCarthy Lawrence Berkeley Labs Berkeley, California	Information Management for Distributed Scientific Data .and
3:00	Refreshments	
3:15	James Dolby San Jose State University San Jose, California	Put the Information in the Database Not the Program .
4:15	Douglas Lenat MCC Corporation Austin, Texas	Relevance of Machine Learning

Data Base Management Conference

SCHEDULE (continued)

Friday, November 2

9:00	Wesley Nicholson and Paula Cowley Battelle-Pacific Northwest Labs Richland, Washington	Managing the Data Analysis Process
10:00	Refreshments	
10:15	Howard Resnikoff Thinking Machines Corporation	Remarks on Parallel Computer Architecture and Information Retrieval
11:15	Francis Narin Computer Horizons Incorporated Cherry Hill, New Jersey	Linking Science, Technology, and Economics Data
12:15	Lunch	
1:30	George Vladutz ISI — Institute for Scientific Information Philadelphia, Pennsylvania	The "World Brain" Today: Scientographic Databases. What are they, How are they created and What are they used for?
2:30	Chester Schuler Central Intelligence Agency	Beyond Storage Retrieval
3:30	Refreshments	

TABLE OF CONTENTS

Section Page

DATABASES FOR STATISTICS

Gio Wiederhold

Department of Medicine and Computer Science

Stanford University

Stanford, California

DATABASES FOR STATISTICS

Gio Wiederhold

Department of Medicine and Computer Science

Stanford University

Introduction

This paper will provide a summary of the current status in areas where statistics use and database technology interact. As such it will introduce and define concepts relevant to other papers presented in the workshop, but it will also illustrate some issues of independent concern. While the fields of database and statistics are both in an advanced state of development, their interaction is still in a primitive stage. In this paper we will concentrate on issues of databases specific for managing data which are subject to statistical analysis.

Basic issues of database research will not be discussed, although some references to the literature will be included. A number of definitions have to be included, since a terminology linking distinct disciplines is likely to be inconsistent. Since the field of databases is new it suffers from terminological problems itself. Some implementation issues have to be presented, since the utility of databases technology is affected by its performance in serving the statistical application tasks.

The motivation for the use of databases is the expectation that large, long-term collections of facts are potential sources of information for decision-making. The data will be obtained from a variety of sources, and be accessible to a variety of users, ranging from regular support functions to ad hoc queries for hypothesis generation, planning, and decision-making. The enabling events for computerization of databases on are the advent of long-term reliable storage in the early 1960s, the capability for multi-user access to computers later in the decade, and the greatly reduced costs of storage and processing in subsequent years. The recent introduction of distributed systems introduces new opportunities of retaining local control while sharing access to data for analysis [Wiederhold'84]. Commercial databases serve a large variety of operational and decision-making functions, and statistical inference is but a part of the system.

For specific studies data may be collected to order. In those cases we typically start with a defined hypothesis and collect only as much data, records and attributes, as required to establish or reject the hypothesis. In those cases general database support is rarely used. Many statistical packages provide file support for this type of data directly.

Definitions

While a database is simply a collection of related data, a DataBase Management System (DBMS) is an integrated set of programs to manipulate and protect the database. The database may be accessed by multiple users and programs simultaneously.

Within the database the data from a database may be organized into distinct files. One file is comprised of a set of similar records. A record in turn is composed of a sequence of fields. This hierarchy implements the divide-and-conquer concept at a storage level over a certain domain, perhaps an institution or a scientific field. At the conceptual level another terminology is appropriate. The database covers the information for the domain, and the concepts of an entity type is represented as a file, an entity instance as a record, and the atomic values as the contents of the fields. Table 1 places the terminology into such categories, and includes corresponding terms used for modeling, to be introduced in a later section of this paper.

Concept	Storage Organization	Modeling term
Enterprise information	Database	Database
Entity type	File	Relation
Entity instance	Record	Tuple
Value type	Field Meta data	Attribute domain
Values or facts	Field	Attribute value
Relationship	Linkage Connection	Join attribute

Table 1: Terminological correspondences.

Many statistical databases are not handled today through database management systems. There are often data storage facilities attached to statistical programming systems, and other statistical databases are managed by their users through *file management functions*. Conceptual and modeling tools are then not emphasized.

The schema:

The essential feature of a database management system is that it uses a machine readable description of the data. This information, also called meta data, is collected within a schema. The schema is then used to control all input and output operations on the database.

The schema serves the long-term and multi-user function of the database. Information about the database is now removed out of the programs which use the database and kept in a central location, where it can be easily managed and shared. No single user can make arbitrary decisions about the organization of the database, but every user can inspect the schema to find out what data is available, and how and where it is stored. This separation of storage and processing control functions is important, because databases have longer lifetimes than the programs, longer than the specific applications which use them, the specific users which interact with them, and certainly a longer life than the computer systems which support them.

During the lifetime a database has to be able to change, since both the enterprise, its environment, and our understanding of the information system changes over time. The schema also mediates the interface between users and the data as the databases changes.

The programs which comprise a database management system (DBMS) manage both the meta data, that is data used to describe the entire database, and the actual data values in the files. The lower-level data management functions may be carried by a distinct subsystem, a file management system.

The data structure:

The data in the database derive their meaning not only from the values they represent, but even more from the relationship with other data elements in the database. A value, say a person's weight, means little if cannot identify that person. Professor Dolby's presentation expands this notion greatly. The DBMS has to support these concepts. Data within a record is related by being part of the description of an entity instance. The schema entry describing the record will provide meta data as: field-name, value-type, value-size, etc. for each component field, and identify the field which labels the entity instance, say social-security-number for a person, and identifies the dependent fields, say the sex, the date-of-birth, and the employer-name. Attributes which are relevant for statistical analysis may also be added. It is useful to document if the encoded values are continuous or discrete, or values are metric or ordinal. Few systems today support these notions fully, although some DBMS's oriented towards data analysis are developing into that direction.

Relationships among entity instances can be represented in a variety of ways. Related entities can be found by the user by specifying procedures based on values in the records. In a personnel file with employees and bosses this means finding the boss's record where bosses.name = employees.boss-name. Whenever such potential relationships are known a priori they should be listed in the schema. For example, information that employee instances can refer to a boss or a school instance can be important to effective use of the database system. If the general validity of a relationship is known, and the instances match, then the system can insert a cross-reference into the person record, so that these related records can be rapidly retrieved.

Since users and application programs refer to data by name, rather than by position, it is now possible to add data fields and relationships for new applications without affecting any prior applications. Fields may be deleted or reassigned only affecting those applications which require the information. Changes in relationships may require a global re-verification of consistency.

The data description languages (DDL) used to describe the data to database management systems are still quite ad hoc and inadequate to define any aspect of data not directly supported by the given database management system. A number of examples are given in Chapter 8 of Wiederhold '83.

The schema, in addition to semantic meta data will also contain information about the files and their organization within the computer hardware. If a schema is used changes of computer equipment should not effect users at all. For these entries in the schema the criteria are not logical correctness but the concerns are security and performance. The best performance is of course achieved by optimal matching of the users' access pattern with physical structures having good locality in terms of access cost. These issues are considered in Chapter 5 of Wiederhold'83. In statistical databases, since access patterns are quite predictable, a very good performance is possible.

Database Models

In order to describe database structures at a higher conceptual level the concept of models has become important. The principal models: relational, hierarchical, and network are all well defined, see for instance Date '81 or Ullman '83.

It is often not well appreciated that models have a variety of functions, and that it is not necessary that the designers, users, or implementors use the same model. Having a similar model for all purposes simplifies the mechanical interfaces in database systems by avoiding mapping functions between the modules. However, limiting the systems to direct mappings will force compromises: the user may have to adapt statistical concepts to the system structure, a structure chosen for efficient implementation. These structures may include directives which are not relevant, and in fact hinder, a clear understanding of the intent of the database. An example of of the latter case is the CODASYL network specification. Performance issues change independent of the users need, and if the user is involved with them, can restrict the adoption of new technology. The issue, of distinguishing accessing concepts from storage implementation, is being addressed in many systems. For instance, recently front-end processors which provide a relational interface for CODASYL systems are being provided by their vendors.

The three models, as well as many other structures, can all be described using primitives of the structural model [Wiederhold '83, Ch. 7]. The components of the structural model are pure relational data structures (relations having tuples to represent entities and entity instances) and three types of connections among them:

Ownership connections
linking lower entity instances which depend on the owner;
the lower level entities disappear when the owner is removed.
(e.g., clinic-visits owned by patients)

Reference connections
linking from instances to higher level abstract referenced entities;
the referenced entities are constrained to exist while refered to.
(e.g., employee refering to a department)

Subset connections
linking from generalization to a subtype;
the subtype instance is removed when the general instance is deleted.
(e.g., employees linked to bosses)

Now we can describe the models as follows:

The relational model does not describe connections.
The hierarchical model only permits single-owner connections.
The CODASYL network model permits multiple owners for one subsidiary entity.
The DAPLEX network model includes the subset connection as well.
The SOCRATE system of the french CII company supports references and single-owners.

The implementations however differ:

A basic relational implementation does not support any connections, but some implementations support reference connections.

The ORACLE relational database management system uses hierarchies for its internal structure and maintains pseudo-owner instances when needed.

CODASYL database management systems permit implementation of reference connections and subset connection through manual control of the standard ownership connections.

The ownership of the TOTAL network implementation permit only one level of owners, but the ownership implementation supports the subset connection as well.

The use of these concepts can help the statistical user to understand the facilities provided by database management system, both externally and internally.

Facilities for Statistics:

After having presented a global summary of database management as now seen in commercial practice, we can investigate the relevance of these systems to tasks in statistics.

Impedance Mismatch:

The most serious conceptual problem in the database-to-statistics interface is that conventional programs are commonly interposed between the database management system, and the statistics packages. These programs will select and retrieve the specific data, store them in some intemediary form, and then invoke the analysis routines.

The problem is that

a) A modern DBMS deals well with sets and subsets of data.

b) The mediating programs deal with individual data values and must tediously extract data elements one-by-one from the DBMS.

c) Statistical analysis programs themselves deal well with sets and subsets of data.

In order to overcome this problem some DBMS include what they consider statistical operators in their system. These are however typically limited to COUNT, MAX, MIN, and AVG, even the SD is rarely included.

An alternative solution is employed by the writers of statistical packages. They add file management capabilities to their systems. In these systems, however, insufficient attention is given to operational issues as input-output, multi-user interaction, and recovery or backup. These systems can not be used operationally, and at best receive a static copy, via a programmed interface, of operational data.

Schema facilities:

The database management system keeps its meta data in a schema. Access to the schema is often limited, so that important type and labeling information is not available to the statistical systems.

It is possible to include in the schema important information for statistical analysis, for instance whether apparently numerical data is really nominal (i.e., defines classes), discrete, or continuous. Such information can be critical both to the human analyst and to automatic analysis control functions, but its use has not been widely exploited. Until such semantic information can be effectively shared by the DBMS and the statistics packages, there will be little motivation to improve the semantic description of data within the DBMS. Without such meta data it will be harder for the DBMS to enforce constraints in data acquisition which are appropriate for eventual statistical analysis.

Data types:

There is rapid progress in database management systems in handling complex data types, for instance 'date-and-time'. Computational functions are now available in many DBMS' for the computation of intervals of dates, their intersection and precedence. Most statistical analyses take a very static view of data and do not deal well with observations made serially over time, especially when they occur at varying intervals. The Time-Oriented Database system (TOD) and its successors are an attempt to address these issues [Wiederhold '75].

File structures:

The typical statistical query addresses few attributes and many record instances. An internal file structure used now by a number of statistics-oriented systems is the transposed structure, where the physical storage brings the same fields of all records together, see Wiederhold'83, Chapt. 4.1. Such structures are included for instance in TOD (Stanford University), ROBOT (Software Sciences, Farnborough, UK), MEDLOG (Information Analysis Corp., Palo Alto, Ca), IMPRESS, RAPID (Statistics, Canada), SC SS, and PICKLE CSA, but are not supported by generalized database management systems.

Complex Structures:

Data analysis is commonly done using flat files, which are easily modeled as relations. Commercial systems recognize more complex relationships. These can be retranslated to flat files by appending much identifying information, but this information is not processed by the statistical programs, and only shuffled around. Since the volume of identifiers can easily exceed by a considerable factor the subject data, more attention should be given to effective handling of this aspect of statistical data management.

Conclusion

The level of understanding of database technology and statistics requirements should permit major progress in this interface in the forthcoming years. Statistical inferencing is an essential part of extracting information from the masses of data being collected as a by-product of modern quantitative management. Relatively little effort is expended into this area, a surprising contrast with the attention given to logical inferencing from databases, see for instance Kerschberg '84.

As database technology and its applications grows there will be increased need for statistical inferencing procedures, which can deal better with the uncertainty inherent in natural world phenomena.

References

Date, C.J. An Introduction to Database Systems, 3rd ed., Addison-Wesley, 1981.

Kerschberg, Larry (ed.). Expert Database Systems, Proc. of a Workshop held Oct. 1984 on Kiawah Island, 2 vols., University of South Carolina, Institute of Information, Management, Technology, and Policy, to be published by Springer Verlag, 1985.

Ullman, J.D. Principles of Database Systems, 2nd ed., Computer Science Press, 1983.

Wiederhold, Gio, James F. Fries, and Stephen Weyl. "Structured Organization of Clinical Databases", Proc. of the 1975 National Computer Conference, AFIPS vol. 44, pp. 479-485.

Wiederhold, Gio, Database Design 2nd ed., McGraw-Hill, 1983.

Wiederhold, Gio, "Databases," Computer Magazine, Vol. 17, No. 10, Oct. 1984, pp. 211-223.

DISCOVERY, CONFIRMATION, AND INCORPORATION OF CAUSAL RELATIONSHIPS FROM A LARGE TIME-ORIENTED CLINICAL DATABASE: THE RX PROJECT

Robert Blum

Stanford University

Stanford, California

DISCOVERY, CONFIRMATION, AND INCORPORATION OF CAUSAL RELATIONSHIPS FROM A LARGE TIME-ORIENTED CLINICAL DATA BASE: THE RX PROJECT

Robert L. Blum

Department of Computer Science

Stanford University

The objectives of the methods and computer implementation presented here are (1) to automate the process of hypothesis generation and exploratory analysis of data in large nonrandomized, time-oriented clinical data bases, (2) to provide knowledgeable assistance in performing studies on large data bases, and (3) to increase the validity of medical knowledge derived from nonprotocol data. The RX computer program consists of a knowledge base (KB), a discovery module, a study module, and a clinical data base. Utilizing techniques from the field of artificial intelligence, the KB contains medical and statistical knowledge hierarchically organized, and is used to assist in the discovery and study of new hypotheses. Confirmed results from the data base are automatically encoded into the KB. The discovery module uses lagged, nonparametric correlations to generate hypotheses. These are then studied in detail by the study module which automatically determines confounding variables and methods for controlling their influence. In determining the confounders of a new hypothesis the study module uses previously "learned" causal relationships. The study module selects a study design and statistical method based on knowledge of confounders and their distribution in the data base. Most studies have used a longitudinal design involving a multiple regression model applied to individual patient records. Data for system development were obtained from the American Rheumatism Association Medical Information System.

1. Information

Every year as computers become more powerful and less expensive, increasing amounts of health care data are recorded on them. Motivation for collecting data routinely into ambulatory and hospital medical record systems comes from all quarters. Health practitioners require this data for clinical management of individual patients. Hospital administrators require it for billing and resource allocation. Government agencies require data for quality of health care assessments. Third party insurers require it for reimbursement. Data bases may also be used for performing clinical research, for assessing the efficacy of new diagnostic and therapeutic modalities, and for the performance of postmarketing drug surveillance.

The various users for data bases may be grouped into two fundamentally distinct categories. The first category pertains to uses that merely require retrieval of a set of data. For example, we may wish to know the names of all patients who had a diastolic blood pressure greater than 100 for more than 6 months and who received no treatment. Uses of medical record systems for patient management, billing, and quality assurance usually fall into this category.

The second use of data bases is for deriving or inferring facts about the world in general. For example, we might request data from a health insurance data base on occupation and hospital diagnoses to determine whether certain occupations are associated with an increased prevalence of heart disease. Here the predominant interest is in generalizing from the data base and only secondarily in the particular values in the data base. The use of data bases for determining causal effects of drugs, for establishing the usefulness of new tests and therapies, or for determining the natural history of diseases falls into this latter category.

The possibility of deriving medical knowledge from data bases is an important reason for establishing them. Given a collection of large, geographically dispersed medical data bases, it is easy to imagine using them for discovering new causal relationships or for confirming hypotheses of interest.

The RX project, as this research project is called, is a prototype system for automating the discovery and confirmation of hypotheses from large clinical data bases. The project was designed to emulate the usual method of discovery and confirmation of medical knowledge that characterizes epidemiological and clinical research. To illustrate this method consider the following hypothetical scenario.

2. Evolution of Empirical Knowledge

Suppose a medical researcher has noticed an interesting effect in a small group of patients, say unusual longevity. He carefully examines those patients' records looking for possible explanatory factors. He discovers that heavy physical exertion associated with occupation and sports is a possible factor in promoting longevity.

Interested in pursuing the hypothesis that heavy physical exertion predisposes to long life, the medical researcher consults with a statistician, and together they design a comprehensive study of this hypothesis. First they analyze the results of the study on their local data base, controlling for factors known to be associated with longevity. Having confirmed the hypothesis on one data base, they proceed to test the hypothesis on many other data bases, modifying the study design to allow for differences in the type and quantity of data.

Having confirmed the hypothesis, they publish the result, and other researchers proceed with further confirmatory studies, attempting to elucidate the mechanism of "exercise effect." When future researchers study other factors that influence longevity, they control for physical activity.

This cycle in which knowledge gradually evolves from data through a succession of increasingly comprehensive studies is illustrated in Fig. 1. At each stage of discovery and confirmation existing medical knowledge is used to design and to interpret the studies.

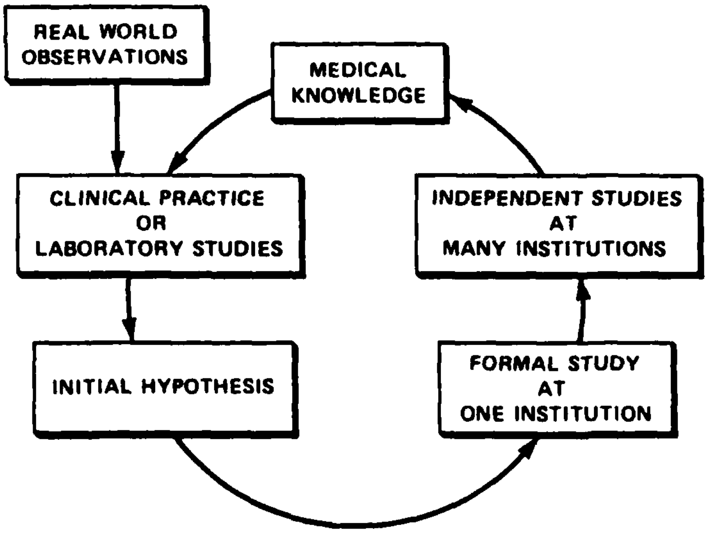

REAL WORLD OBSERVATIONS
MEDICAL KNOWLEDGE
CLINICAL PRACTICE OR LABORATORY STUDIES
INDEPENDENT STUDIES AT MANY INSTITUTIONS
INITIAL HYPOTHESIS
FORMAL STUDY AT ONE INSTITUTION

FIG. 1. The evolution of medical knowledge.

3. The RX Project

It is easy to imagine automating at least parts of the above discovery and confirmation cycle. We obtain our initial hypotheses by selectively combing through a large data base, examining a few patient records guided by prior knowledge. These clues are then studied more comprehensively on the data base as a whole. To design and interpret these studies, medical and statistical knowledge from a computerized knowledge base is used. The final results are incrementally incorporated into the knowledge base, where they can be used in the automated design of future studies.

This describes the RX computer program, a prototype implementation of these ideas. Besides a data base, the RX program consists of four major parts: the discovery module, the study module, a statistical analysis package, and a knowledge base (Fig. 2).

- The discovery module produces hypotheses "A causes B." The hypotheses denote that in a number of individual patient records "A precedes and is correlated with B." Information from the knowledge base is used to guide the formation of initial hypotheses.
- The study module then designs a comprehensive study of the most promising hypotheses. It takes into account information in the knowledge base in order to control for known factors that may have produced a spurious association between the tentative cause and effect. The study module uses statistical knowledge in the knowledge base to design an adequate statistical model of the hypothesis.

- The statistical analysis package is invoked by the study module to test the statistical model. The analysis package accesses relevant data from patient records, and then applies the statistical model to the data. The results are returned to the study module for interpretation.

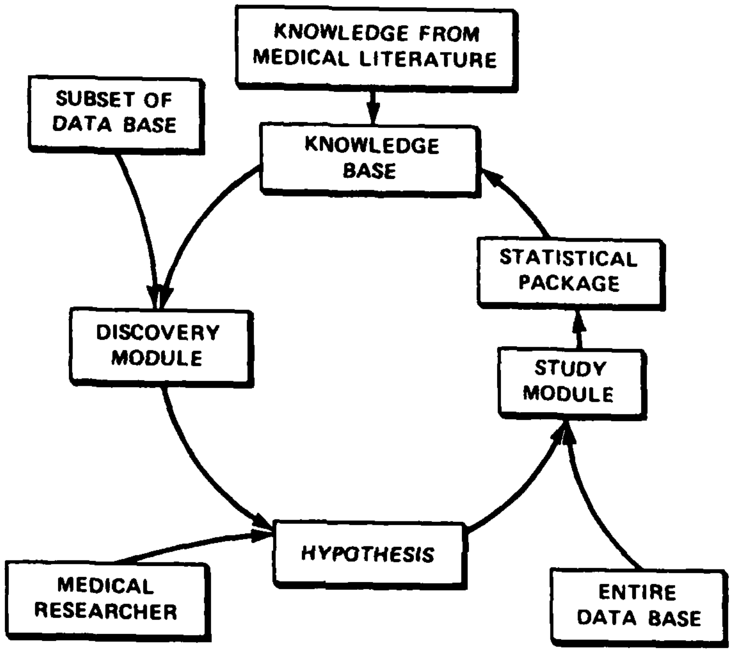


FIG. 2. Discovery and confirmation in RX.

- The knowledge base is used in all phases of hypothesis generation and testing. If the results of a study are medically and statistically significant, they are tentatively incorporated into the knowledge base where they are used to design further studies. Newly incorporated knowledge is appropriately labeled as to source, validity, evidential basis, and so on. As the knowledge base grows, old information is updated.

Currently, the RX program uses only one data base: a subset of the ARAMIS Data Base. Also the extent of medical and statistical knowledge is limited, since the purpose of the research was primarily the development of methodology.

While the program is a prototype, it has been operational since 1979 and has been widely demonstrated. Several interesting medical hypotheses (in varying states of confirmation) have been discovered by the program, including some with little prior supporting evidence.

The objective of this paper is to present an overview of the RX project. Details on statistical methods, modeling of causal relationships, and methods of knowledge representation may be found in Ref. (1).

4. Time-Oriented Data Bases

The general format of a patient record is illustrated in Table 1. Each time a patient is seen in clinic a number of observations are made. These are recorded with the date of observation in the data

Visit number	1	2	3
Date	January 17, 79	June 23, 79	July 1, 79
Knee pain	Severe	Mild	Mild
Fatigue	Moderate	-	Moderate
Temperature	38.5	37.5	36.9
Diagnosis	Systemic lupus		
White blood count	3500	4700	4300
Creatinine clearance	45	-	65
Blood urea nitrogen	36	33	-
Prednisone	30	25	20

Table 1: Hypothetical Time-Oriented Record for One Patient

base. The recorded characteristics of a patient are known as primary attributes or simply attributes. Attributes may be real-valued, rank, categorical, or binary. The term attribute includes all recorded signs and symptoms, lab values, diagnoses, therapy, and functional states.

The defining characteristic of a time-oriented data base is that sequential values for each attribute may be recorded. Note that different attributes may be recorded on different patients, and that the time intervals between values will usually differ. Some attributes may have values that are only sporadically recorded or not at all. In general, the quantity and character of data across patients may vary greatly.

All of the research reported here was done using a subset of the ARAMIS/TOD data base of rheumatology (American Rheumatism Association Medical Information System/Time-Oriented Data Base) collected at Stanford University from 1969 to the present (2, 3). The subset contains the records of 50 patients with severe systemic lupus erythematosus (SLE). The average number of clinic visits for each patient was also 50, and the average length of follow-up was 5 years. Patient records contained 52 attributes.

The size of the data base used in this project, a small sample of the ARAMIS data base, is approximately a half million characters -- much greater than available core storage on our computers after programs have been loaded. Patient records are kept in hash files on disk where they are stored in compressed and transposed format. Indices for each attribute are maintained specifying numbers of values for each patient. Details of data storage and display methods may be found in Ref. (4).

5. Computer Facilities and Languages

Research was performed at two computer facilities at Stanford University: SUMEX/AIM and SCORE. SUMEX/AIM features a DEC dual processor KI-10 running the TENEX operating system. SCORE has a DEC 20/60 running TOPS-20. The ARAMIS data base per se is stored at the Stanford Center for Information Technology on an IBM 370/3033. Data transfer was accomplished by magnetic tape.

All computer programs are written in INTERLISP/(5), a dialect of LISP, a language that is highly suitable for knowledge manipulation. Statistics are performed in IDL (Interactive Data-Analysis Language) (6), discussed later. The RX source code with knowledge base comprises approximately 200 disk pages of 512 words each.

6. The Knowledge Base

While the prospect of using clinical data bases to discover or to confirm medical hypotheses is tantalizing, there are formidable problems in making inferences from nonrandomized, non-protocol data. These include numerous forms of treatment and surveillance bias, poor adjustment for covariates, inadequate specification of patient subsets, and improper use of statistical analysis (7-9). The use of nonrandomized data for clinical inference demands more stringent data analysis, study designs of greater sophistication, and more thoughtful interpretation than does the use of data gathered in a randomized trial.

The leitmotif of the RX project is that derivation of new knowledge from data bases can best be performed by integrating existing knowledge of relevant parts of medicine and statistics into the medical information system. During the evolution of a medical hypothesis, as was illustrated, existing medical knowledge comes into play at every stage.

In the RX computer program the medical knowledge base determines the operation of the discovery module, plays a pivotal role in the creation of subsequent studies in the study module, and finally serves as a repository for newly created knowledge. The medical knowledge base grows by automatically incorporating new knowledge into itself. Hence, it must be designed in such a way that relationships derived from the data base can be translated into the same machine-readable form as knowledge entered from the medical literature by a researcher. In any case knowledge relevant to a study must be automatically accessible.

The main data structure of RXs knowledge base (KB) is a tree representing a taxonomy of relevant aspects of medicine and statistics. Each object in the tree is represented as a schema containing an arbitrary number of property: value pairs. The RX KB contains approximately 250 schemata pertaining to medicine, 50 pertaining to statistics, and 50 system schemata. The medical knowledge in the RX KB covers only a small portion of what is known about systemic lupus erythematosus and some areas of general medicine. The present KB is merely a test vehicle; its size is 50 disk pages or 120,000 bytes.

6.1. Medical Knowledge

The medical knowledge base is a subtree of the KB distinct from the statistical knowledge base. Its first-order subtrees are *states* and *actions*, which in turn are broken down into *signs, symptoms, lab findings, diseases* and into *drugs, surgery*, and *physical therapy*. The categories of diseases and other entities follow the conventional nosology based upon organ systems and/or pathology found in any standard textbook of medicine (10). I shall occasionally refer to each of the objects in the medical KB as a *node* and to the information stored at each node as its *schema*.

The schema for each object is represented as a collection of property: value pairs called a *property list*. In general the objects in the KB are either primary attributes in the data base or are *derived variables*, that is, objects whose values must be derived from primary data. The properties in the schema of the object may be grouped into the following categories: *data base schema properties, hierarchical relationship properties, properties describing the definition of an object and its intrinsic properties, and properties describing cause/effect relationships to other objects*.

6.1.1. Data Base Schema Properties

Each of the attributes in the clinical data base is represented by a schema in the KB describing its units of measurement, how its values are stored, and so on. This kind of schema is typical of most data bases today (11). As an example, part of the schema for the attribute *hemoglobin* appears below.

Hemoglobin

---

attribute-type: point-event
value type: real {i.e., a real-valued number}
range: 0 < value < 25
significance: 0.1 {i.e., values are rounded to the nearest 0.1}
units: grams per deciliter

6.1.2. Hierarchical Relationship Properties

Two properties are used to store the position of an object in the medical hierarchy: specialization and generalization, abbreviated spec and genl as below.

Inheritance mechanisms (12) are used by the study module as a means for exploiting the knowledge implicit in the hierarchy. For example, in the course of a study, if the expected duration of klebsiella pneumonia was required to construct a statistical model, then a default value might be inherited from the schema for pneumonia.

Respiratory diseases

genl: All categories of disease
spec: Pneumonia, asthma, emphysema

Pneumonia
genl: Respiratory dis.
spec: Pneumoncoccal pn.
Klebsiella pn.

Asthma
genl: Repiratory dis.
spec: Allergic Asthma
Intrinsic asthma

Emphysema
genl: Repiratory dis.
spec: CO_2 retention

6.1.3. Properties Pertaining to the Definition and Intrinsic Characteristics of an Object

If an object is a primary data base attribute such as hemoglobin, then no definition is required, at least not from a standpoint of deriving values for it. Values for hemoglobin are simply those in the data base.

On the other hand, it the values for an object are derived from primary attributes, the specification of the means for derivation must be recorded in the KB. That is the definition of the object. The didactic example below shows a definition for pneumonia.

Pneumonia

---

defintion: Temperature > 38°C
and WBC > 10,000 cells/mm^3
and Chest X-ray = lobar infiltrate

In the RX KB the specification and use of definitions are far more complicated than is suggested by this example. Recall that data-base attributes are time-oriented with nonuniform time intervals and frequently missing values. Hence, definitions of derived objects must contain time-dependent predicates and mechanisms for handling sporadic values. Definitions can also refer to other derived objects. The temporal characteristics of an object may be specified using other properties in the schema: expected duration, carryover, onset-delay, and so on. These parameters are used by the time-dependent predicates when definitions for objects are evaluated.

6.1.4. Properties Specifying Causal Relationships to Other Objects

The final class of properties are those specifying the causal relationships of an object to other objects. In RX all causal relationships are stored using two properties: effects and affected-by. The effects property records a list of those objects directly affected by the object. The affected-by property contains a list of objects that directly affect it. Additionally, the detailed characteristics of the causal relationship between a pair of objects is stored on the affected-by property. The resulting causal model is a directed cyclic graph; that is, the representation allows for the possibility that A causes B causes A.

Besides the simple fact that A may affect B, each causal relationship is represented by a set of features as below.

<intensity, frequency, direction, setting, functional form, validity, evidence>

Briefly, these take the following form when both the cause and effect are real valued.

- intensity: the expected change in the effect given a change in the cause, expressed as an unstandardized regression coefficient,
- frequency: the distribution of the effect across patients, expressed as deciles of the expected effect given a "strong" change in the causal variable,

- direction: increase or decrease,
- setting: the clinical circumstances specifically included or excluded or excluded from the study, expressed as a Boolean with time-dependent predicates,
- functional form: the complete statistical model used to study the relationship, expressed in machine-readable form,
- validity: a 1 to 10 scale distinguishing tentative associations from widely confirmed causal relationships,
- evidence: a summary of the study performed by the Study Module, including patient IDs, methods, and intermediate results.

The entire causal relationship is machine readable. This enables it to be used automatically by the study module during subsequent studies. The causal relationships in the KB can also be interactively displayed in a variety of forms. All paths connecting two nodes may be displayed, or the details of a particular causal relationship: its mathematical form, the evidence supporting it, or its distribution across patients. In the example below the effects of prednisone have been displayed. The verbs and adverbs in the phrases are supplied by a lexicon during machine translation.

PREDNISONE, at a level of 30 mg/day {modal effects}

usually increases CHOLESTREROL by 50 to 130 mg/dl,
usually increases WEIGHT by 3 to 7 kg,
regularly attenuates NEPHROTIC SYNDROME by 1 to 2 g protein/24 hr.
regularly attenuates GLOMERULONEPHRITIS by 10 to 30% activity,
regularly decreases EOSINOPHILS by 2 to 3% of WBC,
commonly decreases ANTI-DNA by 50 to 90% activity,
occasionally increases GLUCOSE by 20 to 100 mg/dl.

7. The Discovery Module

The general methodology used by RX to discover and then to study causal relationships is known as a "generate and test" algorithm. Briefly, the discovery module proposes causal links based on a test for strength of association and time precedence. After a number of tentative links have been added, the study module performs and exhaustive study of them in the same order in which they were added. In the course of this study many tentative links well be removed, and the remaining ones will be labeled with detailed information on the respective relationships. After a link has been incorporated into the model, it may be used to refine the study of further links.

7.1. An Operational Definition of Causality

Underlying the discovery module and the study module is the following operational definition of causality. A is said to cause B if over repeated observations (1) A generally precedes B, (2) the intensity of A is correlated with the intensity of B, and (3) there is no known third variable C responsible for the correlation.

These properties are the foundation of the RX algorithm. I will refer to these properties as (1) time precedence, (2) covariation or association, and (3) nonspuriousness (13, 14).

Causality can never be proven using observational data. The persuasiveness of a given demonstration simply depends on the extent to which the three properties have been shown.

7.2. Methodology of the Discovery Module

The function of the discovery module is to find candidate causal relationships. The discovery module exploits only the first two properties of causal relationships to do this: time precedence and covariation.

The discovery module considers all pairs of variables {A, B}, where A and B are either primary attributes in the data base or are derivable from primary attributes. It attempts to determine whether the data suggest that A causes B, B causes A, both, or neither. The output of the discovery module is an ordered list of hypotheses. A researcher may designate which potential causes and effects are of interest. For example, certain drugs and diseases might be tagged as being of interest in exploration. The algorithm is intrinsically slow, $O(n^2)$, where n is the number of variables; however, it makes up for this inefficiency by its sensitivity and the speed with which simple correlations can be performed.

A pairwise algorithm was chosen for the discovery module after months of experimentation with multivariate methods. The latter cannot be applied to data of the type recorded in the ARAMIS data base without extensive loss of information. The reason is that values are only sporadically recorded and patients differ widely on covariates. The general philosophy in all RX procedures in either the discovery module or the study module is to analyze data only within individual patient records. That is, data in two patient records are never combined before statistical analysis. The computational expense incurred by analyzing individual patient records will decrease markedly when multi-cpu machines become standard.

The basic algorithm uses a sliding nonparametric correlation performed on data from an individual patient's record. The principle underlying a lagged correlation is illustrated in Fig. 3. Given a tentative cause A and effect B, the basic tool for uncovering a casual relationship is the Spearman correlation coefficient $r_s(A, B, \tau)$, where τ is the time delay used in computing the correlation.

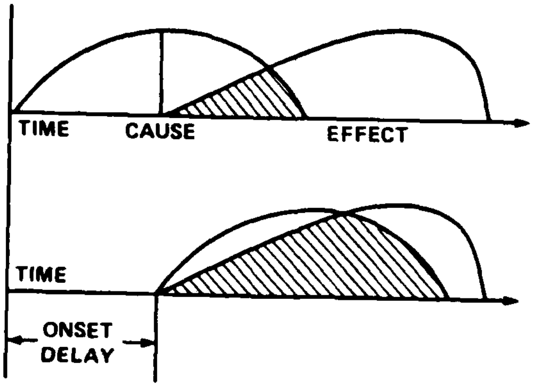


FIG. 3. The principle underlying lagged correlation.

7.2.1. Selection of Patients for Correlation

In the discovery module only a sample of the patients records are analyzed. The sampling procedure uses a precomputed index called a records list associated with every variable in the data base. The records list is a sorted list of the form ((patient$_1$, n_1), (patient$_2$, n_2), ..., (patient$_m$, n_m)). The list identifies patients in descending order by their number of recorded values for the variable. That is, patient$_1$ has n_1 measurements of the variable, and so on.

The sample of records that are analyzed for a given pair of variables $\{A,B\}$ is the sample $P^*_{\{A,B\}}$, where this is the set with the largest number of pairs of measurements of A and B. Let K denote the number of pairs in the set $P^*_{\{A,B\}}$. In experimental trials of the discovery module, K was set to 10.

The advantage of choosing the sample to be those patients with the most data on A and B is that "one might as well look where the looking is best." If a relationship exists between A and B, then it will be easiest to detect in patients with lots of data on A and B. This heuristic is particularly valid in medical data when variables are more apt to be recorded when they are abnormal. Therefore, the frequency of observation tends to be correlated with the variance of the variable.

Correlations for the records in $P^*_{\{A,B\}}$ are computed as

```
for each record in $P^*_{\{A,B\}}$ collect
    [for each $\tau$ in $T^*$ collect $r_s(A, B, \tau)$] .
```

The collect operator denotes assembling a set composed of the value of each iterand. The time delays in T* over which the correlations are performed are based on information from the knowledge base. That is, the algorithm makes use of prior information on the expected time delays of broad classes of causes and effects.

7.2.2. Combining Correlations Across Patients

That various correlations within and across patient records are based upon different numbers of measurements poses a difficulty in combining them. Given equal correlations, we would like to assign more weight to records with more data. Using the p value of the correlation achieves this and also facilitates combining correlations.

The p values from the above procedure may be diagrammed as

$$\begin{array}{lcccc} & \tau_1 & \tau_2 & \cdots & \tau_q \\ \text{patient}_1 & p_{1,1} & p_{1,2} & & p_{1,q} \\ \text{patient}_2 & p_{2,1} & p_{2,2} & & p_{2,q} \\ \vdots & & & & \\ \text{patient}_K & p_{K,1} & p_{K,2} & \cdots & p_{K,q} \end{array}$$

Here $p_{i,j}$ denotes the p value on the ith time delay. By the method of Fisher, the p values may be combined to form an overall score s for each time delay τ_j:

$$s(A,\ B,\ \tau_j,\ P^*_{\{A,B\}}) = -\ 2\Sigma\ \log(p_{i,\tau_j})$$

where the sum is over all patient records in $P^*_{\{A,B\}}$. It can be shown (15) that the scores s are distributed as χ^2 on 2p degrees of freedom. Since the distribution of the scores is known, their statistical significance may be calculated. Because of autocorrelation, the differences between scores determined at different time lags may not be distributed χ^2. However, the significances are not taken literally by the discovery module, but are merely used to rank the hypotheses in terms of promise.

If the difference between the forward and backward sets of scores is large, a strong time precedence of association is implied. Since time precedence is not a sufficient condition for causality, spurious associations may also be reported as significant.

The output of the discovery module is a list of dyadic relations ranked in descending order by strength of unidirectionality of association. The algorithm has proven to be a sensitive, if nonspecific, detector of causal relationships, and is usually capable of accurately discriminating time precedence and determining approximate onset delay.

In the discovery module, only the properties of time precedence and covariation are used in a blind search for clues to causal relationships. Included in its output are many spurious relationships. The objective of the study module is to eliminate those relationships and to carefully examine those that remain in order to detail their characteristics and to store them in the KB.

8. The Study Module

The study module is the core of the RX algorithm. It takes as input a causal hypothesis gotten either from the discovery module or interactively from a researcher. It then generates a medically and statistically plausible model of the hypothesis, which it analyzes on appropriate data from the data base.

The study module is patterned after a sequence of steps usually undertaken by designers of large clinical studies. Its design may be considered an exercise in artificial intelligence insofar as it emulates human expertise in this area. There are at least six persons whose knowledge is brought to bear in designing, executing, reporting, and disseminating a large database study. We may think of the *data-base research team* as consisting of a doctor, a statistician, an archivist, a data analyst, a technical writer, and a medical librarian. The study module, in conjunction with the knowledge base (KB), emulates part of their expertise. The steps in the study module appear in Table 2.

1. Parse the hypothesis.
2. Determine the feasibility of the study on the database.
3. Select confounding variables and causal dominators.
4. Select methods for controlling the causal dominators.
5. Determine proxy variables.
6. Determine eligibility criteria.
7. Create a statistical model.
   (a) Select an overall study design.
   (b) Select statistical methods.
   (c) Format the appropriate database acess functions.
8. Run the study.
   (a) Fetch the appropriate data from eligible patient records.
   (b) Perform a stistical analysis of each patient's record.
   (c) Combine the results across patients.
9. Interpret the results to determine signficance.
10. Incorporate the results into the knowledge base.

Table 2: Steps Performed by the Study Module

8.1. Determination of Feasibility of Study

The study module may be operated automatically in batch mode, or it may be run interactively, enabling a researcher to modify the evolving study design. In this presentation we shall assume that it is being run interactively. Throughout this section we shall use as an example the hypothesis that the steroid drug prednisone elevates serum cholesterol.

The first general task of the study module or of the "data-base research team" is to determine whether a particular study is feasible given the knowledge and the data available. The first step is the recognition by the program of the terms used in the hypothesis.

Suppose a researcher enters the hypothesis prednisone elevates cholesterol. A top-down parser is applied to this input string. The pattern that matches is <variable relationship variable> where variable may be any primary attribute or derived variable in the medical KB. As the parser matches the tokens in the input, it determines their classification in the KB.

> Prednisone is a known concept.
> It is classified as a Steroid which is a Drug which is an Action.
>
> Elevates is a known concept.
> It is classified as a Relationship
>
> Cholesterol is a known concept.
> It is classified as a Chemistry which is a Lab-Value which is a State.

The classification are simply determined by following the generalization pointers in the knowledge tree. The classification of each variable is not only of interest to the user but facilitates the inheritance mechanisms discussed above. For example, properties of the class steroids may be inherited by the drug prednisone, if they are needed in the course of the study.

To study the relationship between prednisone and cholesterol, both variables must have been recorded in some patient records. Hence, the program next examines the intersection of their records lists.

Cholesterol

records: ((P78 32)(P118 25) ••• P967 1))

The list here denotes that patient 78 had 32 recorded values for cholesterol, patient 118 had 25 values, and so on.

8.2. Confounding Variables and Causal Dominators

The principal objective of the study module is the demonstration of nonspuriousness. In any observational drug study, as in the current one, the possibility must always be addressed that the effect of interest was caused by the disease for which the drug was given rather than by the drug itself. The first step in demonstrating nonspuriousness is in identifying the set of possible confounding variables.

A confounding variable is any node C that may cause a clinically significant effect on both the causal node A and the effect node B in our hypothesis. The "clinical significance" of a given change in a variable is determined by a prior partitioning of that variable's range. Every real-valued object in the knowledge base has stored in its schema a partition list that divides its range into clinically significant regions.

Let C be the set of known confounders. The determination of C involves tracing the directed graph in the KB starting from A and B.

C = intersection[antecedents (A), antecedents (B)]

where the list antecedents (A) is the set of nodes that may produce a clinically significant effect on A. The antecedents set of a node is calculated by traversing the causal network in the KB. In the current example, the set C is determined to be {ketoacidosis, hepatitis, glomerulonephritis, nephrotic syndrome}.

Having determined the variables in C, the program displays the causal paths connecting them to A and B. The paths for glomerulonephritis appear below. The intensities of intermediate nodes are calculated using the regression coefficients stored in sequential causal relationships.

Glomerulonephritis {50% activity} is treated by Prednisone {30 mg/day},

Glomerulonephritis can cause Nephrotic Syndrome {4g proteinuria/24 hr} which is trated by Prednisone {20 mg/day},

Glomerulonephritis can cause Nephrotic Syndrome {4g proteinuria/24 hr} which increases Cholesterol {65 mg/dl}.

8.3. Causal Dominators

To increase statistical power and stability of estimation it is usually desirable to control for as few confounding variables as possible. Since the set C in any real study is apt to be quite large, it is desirable to control for only the essentials. The set of causal dominators C* is the smallest subset of C through which all known causal influences on both A and B flow.

The set of causal dominators C* is determined in the present computer program by the following algorithm. Assume we are interested in determining whether A causes B. Let us designate by P the set of proximate causes of B; that is, P is simply the set of nodes on B's affected-by list. We first check to determine whether any nodes in P can reach A, i.e., may also causally influence A, however indirectly. Any of those nodes in P that can reach A are appended to the set C* of causal dominators, which is initially empty. Call this set of nodes P_1. Then the nodes P_1 are blocked by placing flags on them. This prevents flow through them on subsequent iterations. Next, consider the set of nodes $P_2 = P-P_1$, and generate the set of all proximate causes of the nodes in P_2. Call this set Q. If we now assign P = Q and iterate the above sequence, the set of causal dominators C* is generated. The algorithm is admittedly inefficient, but adequate for the size of networks with which we have dealt. In the current example, glomerulonephritis is deleted from the confounders since its confounding influence is entirely through nephrotic syndrome.

8.4. Controlling Other Variables

8.4.1. Variables Related to the Cause

Suppose prednisone affects cholesterol in some fashion; it is possible that related drugs may also affect cholesterol. We may also want to remove their influence by controlling them. Generally, we would like the program to suggest to us variables related to the cause, since they may also be confounders. These variables may not be in the set C, since causal paths between them and the effect may be unknown.

To select this set of variables related to the causal variable, the program uses the hierarchical structure of the KB. For example, since *prednisone* is one of the *steroids*, RX controls for the other steroids. These are the *siblings* (prednisone) = {dexamethasone ACTH}: nodes in the same class, *steroids*.

8.5. Determination of Methods for Controlling Confounding Variables

Three general methods are used by RX to control confounding variables: (1) eliminate entire patient records, (2) eliminate time intervals containing confounding events, and (3) control statistically for the presence of the confounder. Eliminating patient records is always the safest and most intellectually reassuring. With stastical control, doubt always remains as to whether the confounder has been entirely eliminated. When eliminating time intervals, the possibility that the confounding influence extends beyond the interval is always possible. On the other hand, eliminating patient records is the strategy most wasteful of data. There may be too few records left to analyze, or the generalizability of the result may be diminished.

To determine which method to use for each confounder, some decision criteria must be used. In making this decision and others discussed later, the study module uses decision criteria stored in the KB in the form of *production rules*.

8.6. Production Rules

Production rules have been widely used in artificial intelligence research to store domain knowledge (16, 17). A production rule is an if/then rule consisting of a premise and conclusion.

The rule below is stored with other similar rules in the schema for control methods. To choose a control strategy, the rules are exhaustively invoked. Some rules may be used to resolve conflicts, if more than one control method is suggested.

```
IF        the number of patients affected by a variable
              is a small percentage of the number of
              patients in the study,
AND       the variable is present throughout those records,
THEN      eliminate those records from the study.
```

The premise and conclusion of each production rule consist of a few lines of machine-readable code. In some systems (17), the code may be mechanically translated into English upon request. To avoid the attendant complexity and to improve the quality of translation, the RX KB simple stores an English translation of each production rule.

In writing programs that use much domain knowledge, it is advantageous to separate the specific knowledge from the general algorithms that use it. Production rules are one method for achieving this modularity. The advantages are that (1) knowledge is more easily examined and updated, (2) dependencies among the knowledge are more easily discovered, and (3) the homogeneous format lends itself to machine translation.

8.7. Controlling Confounders

To determine how a particular confounder is to be controlled, the following information is first determined: N, the number of patient records in the study; %records, the fraction of records affected by the confounder; and %visits, the average fraction of visits affected. Each of these parameters is calculated using the information in the records list for each confounding variable.

If %records or %visits are low, then either records or time intervals may be eliminated. The rules tend to favor the elimination of records if N is high. Only if N is low and %records or %visits is high is statistical control of the confounder considered.

While the program is running the user may request a display of the rules that determined the choice of strategy. The user, as always, may override the decision made by the program.

In the prednisone/cholesterol study the program makes the following selections.

Dexamethasone	No control needed, since no values were recorded in the data base
ACTH	No control needed
Nephrotic Syndrome	Control statistically using albumin as a proxv
Hepatitis	Eliminate affected time intervals
Ketoacidosis	Eliminate affected time intervals

8.8. Choice of Study Design and Statistical Method

Both the study design and the statistical method are selected using decision criteria stored in production rules in the KB. The choice of study design in the present system is simply a choice between a cross-sectional versus a longitudinal design. In a cross-sectional design each variable is sampled once in a patient's record; in a longitudinal design variables are repeatedly sampled over time. The longitudinal study design has the advantage of making use of temporal information and multiple observations of variables within individual patient records. A cross-sectional design is only chosen when a longitudinal design is not feasible.

The selection of a particular statistical method uses knowledge encoded in a hierarchically organized, statistical knowledge base. The organization follows the conventional classification as in Ref. (18) or (19).

On the property list of each node in the tree is an *objectives*, a *prerequisites*, and an *assumptions* property. The *objectives* property describes the goals of the method. The *prerequisites* property describes the conditions that must hold for the method to be mechanically applied. The *assumptions* property describes the assumptions that must hold for the result to be valid.

Multiple regression

---

objectives: linear model

prerequisites:

one dependent variable

two or more independent variables

measurement level of dependent variable = real valued

measurement level of independent variables = real valued

number of observations > 1 + number of independent variables

assumptions:

independent and identically distributed errors

normally distributed errors

linear and additive effects

An example of the schema for multiple regression appears above. The schema stores not only the English text but the equivalent machine-executable code.

To select a statistical method the objectives and prerequisites properties must satisfy the constraints of the study. The tree structure of the KB is used to prune limbs that are not applicable. When there is more than one applicable method, production rules at intermediate nodes arbitrate among methods. The present program does not determine whether the assumptions of a method have been fulfilled; they are merely displayed. It does make available tables and plots of residuals, however, so that the assumptions can be manually checked.

The present version of this robot statistician is rudimentary. Each of the nodes in the statistical KB contains about as much knowledge as is shown for multiple regression. No knowledge or methods are present for critically analyzing a fitted model or for revising the model. The current emphasis is simply in selecting a method that may be mechanically applied.

8.9. Composition of Data Base Access Functions

In order to apply the selected analytical methods to the appropriate data, the data must be sampled from patient records at times that reflect the time delays inherent in the underlying processes. These time parameters are obtained by the study module from information in the KB.

For the longitudinal design in the present example the following model is created;

$$\Delta\text{cholesterol} = \beta_0 + \beta_1 \, \Delta\text{albumin} + \beta_2 \, \Delta \, \log(\text{prednisone})$$

where

$$\Delta\text{cholesterol} = \text{cholesterol}(t) - \text{cholesterol}(t_{\text{pchol}});$$

$$\Delta\text{albumin} = \text{albumin}(t-\tau_{NS}) - \text{albumin}(t_{\text{pchol}} - \tau_{NS});$$

and

$$\Delta \, \log(\text{prednisone}) = \log[\text{prednisone}(t - \tau_{\text{pred}})] - \log[\text{prednisone}(t_{\text{pchol}} - \tau_{\text{pred}})] \, .$$

The time t_{pchol} denotes the time of measurement of the cholesterol previous to the present one, and τ_{NS} denotes the estimated delay from the start of nephrotic syndrome to the establishment of a steady state for cholesterol. The symbol τ_{pred} is the analogous onset-delay for prednisone. No values are sampled during episodes of hepatitis or ketoacidosis. Some of the time relationships that might be seen in one patient's record are illustrated in Fig. 4.

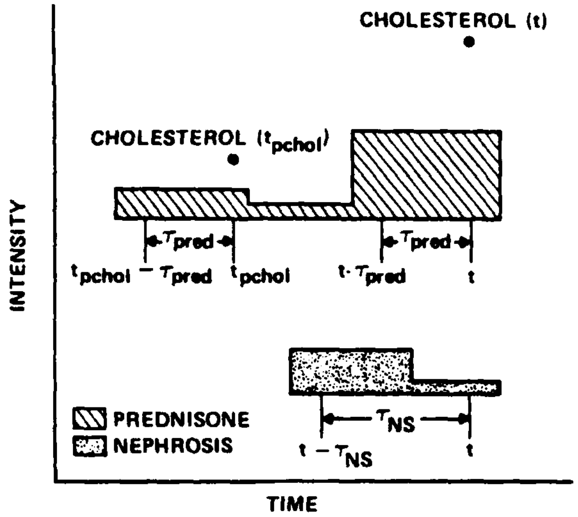


FIG. 4. Time relationships in prednisone: cholesterol study.

Next, the mathematical model must be translated into the appropriate data-base access functions. The function create-access-functions uses information in the schemata for the variables in the model to format the appropriate access functions. For example, the values for the onset-delays and the need for the log transform are retrieved from the schemata for nephrotic syndrome and prednisone. The estimated time delay for the effect of prednisone on cholesterol is obtained from the discovery module.

8.10. Determination of Eligibility Criteria

All patients in a data base may not be eligible for a particular study. Eligibility criteria in the current example are automatically formatted based upon the number of relevant observations in a patient's record and the within-patient variance in the causal variable.

The study design cannot be executed on patient records in which there are less than four sets of observations (= 1 degree of freedom for the mean + 2 df for Δalbumin and for Δprednisone). Furthermore, patient records are excluded in which the coefficient of variation in log prednisone is below threshold.

8.11. Statistical Analysis: Fitting the Model

Until July 1980, all statistical analyses were performed using SPSS (20) as a subroutine; however, this incurred the inefficiency of having to write and read files in formats intended for human usage. Currently all statistical analysis is performed using IDL (6). Written in INTERLISP, IDL makes available fast numerical computation, matrix manipulation, and a variety of primitive operators for statistical computation.

Most of our studies are sufficiently large that statistical analysis requires use of a separate core image (separate job). The study module writes the study design to disk, then calls IDL. IDL reads the study design, executes it, writes the results to disk, then calls the study module.

8.11.1 Longitudinal Design Using Weighted Multiple Regressions

The method of analysis that we have most extensively developed combines the results of separate multiple regression analyses performed on individual patients. Recall that individual patient records differ in quantity of data and greatly vary on covariates. By analyzing each patient's record separately, we can determine the distribution of an effect across patients and obtain information as to why some patient's exhibit an effect and others do not.

Naturally, we are interested in knowing whether a given causal relationship is statistically significant in the study sample as a whole. The analysis of significance is complicated by the fact that patients have widely varying amounts of data. Intuitively, one would like to weight most heavily those patients in whom a relationship has been most precisely determined, e.g., the patients with the most data; however, these patients may be unrepresentative.

The approach we use is a mixed model. The regression coefficient for each patient is weighted by the inverse of its variance. The mathematical justification for this procedure lies beyond the scope of this paper but may be found in Ref. (1). When there is a large variation in the effect across patients, perfect precision on any one patient is of little advantage, and all patients are weighted nearly equally. When across-patient variation is small, weighting be precision is more appropriate, and the weights diverge.

8.12. Interpretation of Results

The final result of the longitudinal design is an estimate of β, the unstandardized regression coefficient of the effect on the cause, and $var(\beta)$, its variance. The ratio $\beta/[var(\beta)]^{.5}$ is approximately distributed as a t statistic on $n - 1$ degrees of freedom, there n is the number of patients in the study. A two-sided p value is calculated using the t statistic.

Presently, the interpretation of the results of a study depend only on the magnitudes of β and its corresponding p value. A significant p value does not necessarily mean the result is medically significant. Even an inconsequentially small change in the effect will become significant at a given p value, if the number of patients is large enough. The program for interpretation uses the following heuristic. If β is large, then for a given p value, it assigns a higher validity to the result than if β is small.

The clinical significance of β is determined by the magnitude of its expected influence on the effect variable in the study. This is illustrated in Table 3, which shows the expected distribution of cholesterol given prednisone at 30 mg/day.

Recall that the validity score is a component of every causal relationship stored in the KB. The validity score is measured on a scale from 1 to 10 summarizing the state of proof of a relationship. The highest score that a study based on a single nonrandomized data base can achieve is 6. Higher scores can be obtained only from replicated studies, the highest scores requiring experimental manipulation and known mechanism of action. A score of 6 means that "strong correlation and time relationship have been demonstrated after known covariates have been controlled in a single data-base study."

The discovery module populates the KB with causal links of validity between 1 and 3. The study module overwrites the links that it explores, assigning to those that it confirms scores between 4 and 6.

A statistician or researcher might choose to pursue a given study further asking, "have the confounding variables in C* been adequately controlled?" "Are the residuals in each of the regressions independent and identically distributed?" "What accounts for the differences among patients?" A researcher can pursue these questions interactively in RX, incrementally improving the mathematical model (21); however, the automation of this kind of inquiry will require building much greater knowledge into the "robot statistician."

Range of cholesterol	Percentage of patients	Magnitude of change
100-150	0	Extreme-
150-195	0	Strong-
195-210	0	Moderate-
210-225	0	Weak-
225-230	0	Equivocal-
230-235	0	Equivocal+
235-250	0	Weak+
250-280	10	Moderate+
280-360	82	Strong+
360-700	8	Extreme+

[a] Distribution across patients of cholesterol (mg/dl), given a baseline value of 230 mg/dl and given a change in prednisone from 0 to 30 mg/day.

Table 3: Distribution of the Pednisone/Cholesterol Effect Across Patients[a]

9. Medical Results

The medical results reported here were generated by running the discovery module and then the study module on a sample database containing the records of 50 patients with systematic lupus erythematosus (SLE). Many patients had multisystem involvement including glomerulonephritis and nephrotic syndrome.

The effects that were confirmed by the study module for the steroid drug prednisone are shown in Table 4. To illustrate the interpretation of Table 4, the second row of the table means that prednisone is thought to cause an increase (+) in cholesterol, that the time delay is "acute" (less than one average intervisit interval), and that the effect is highly statistically significant 9p = .0001). The study module automatically incorporated these new links and details of the studies into the knowledge base in the format discussed above.

	Direction	Onset-delay	p Value
Weight	+	Chronic	< .0001
Cholesterol	+	Acute	.0001
WBC	+	Acute	.0004
Neutrophils (%)	+	Acute	.003
Lymphs (%)	-	Acute	.003
BP-diastolic	+	Acute	.004
Glucose	+	Acute	.007
Hemoglobin	+	Chronic	.009
Wintrobe ESR	-	Chronic	.01
Platelets	+	Acute	.02
Temperature	-	Chronic	.05
Anti-DNA	-	Chronic	.08
Eosinophils (%)	-	Acute	.15
Urine-RBCs	-	Chronic	.17
Creatinine	-	Chronic	.19

Table 4: Effects of Prednisone

Almost all of the acute effects appearing in the table have been extensively confirmed in the medical literature. The effect of prednisone on cholesterol, strongly supported by this study, has been reported only a few times previously. No previous study has recorded the reproducibility of the effect over time or the interpatient variability as was done here.

The chronic effects of prednisone shown in Table 4 are those appearing in a setting of severe SLE. Literature confirmation of these effects has been scant. Because of small numbers of patients, the chronic effects shown here must be further studied. Tables of other empirical results and a discussion of the statistical models used in these studies may be found in Ref. (1).

10. Summary

The methods described here emanate from a small set of operational properties of causal relationships. The discovery module uses a nonparametric method for producing a ranked list of causal hypotheses based on strength of time precedence and association. The study module uses a consensual causal model stored in a knowledge base to determine all known confounding variables and to determine appropriate methods of adjusting for them. The statistical model of the tentative causal relationship is then applied to a set of data. If the results indicate that a relationship is significant after controlling for confounding influences, then a new relationship is incorporated into the KB. Subsequent studies may make use of this new link.

All components of the study module can be used in an interactive mode to enable a researcher more control in determining the course of the study. For example, the causal model stored in the KB can be queried interactively of changed in the course of a study as new information becomes available. All phases of the statistical analysis can also be interactively modified.

Any methodology that draws causal inferences based on nonrandomized data is subject to an important limitation: unknown covariates cannot be controlled. The strength of the knowledge base lies in its comprehensiveness, but even so, it cannot guarantee nonspuriousness. Any single study, particularly one using nonrandomized data, must be viewed skeptically. For this reason, the most conclusive causal relationships that RX discovers are always assigned a modest validity. Only through repeated studies, particularly through experimental manipulation of the causal variable, can a given result become more definitive.

Acknowledgments

I am grateful to Guy Kraines, Kent Bailey, and Byron William Brown for their assistance with the statistical models, to Gio Wiederhold for project administration and guidance, to Beau Shiel and Ronald Kaplan for their assistance with IDL, and to James Fries, Alison Harlow, and James Standish for assistance in obtaining clinical data.

Funding for this research was provided by the National Center for Health Services Research through Grant HS-03650, by the National Library of Medicine through Grant LM-03370, and by the Pharmaceutical

Manufacturers Association Foundation. Computation facilities were provided by SUMEX-AIM through National Institutes of Health Grant RR-00785 from the Biotechnology Resources Program. Clinical data were obtained from the American Rheumatism Association Medical Information System. The project is continuing under the sponsorship of NCHSR Grant HS-04389.

Reprinted with permission from *Computers And Biomedical Research*, Volume 15, (1982), pp. 164-187, Academic Press.

Bibliography

1. Blum, R.L. Discovery and representation of causal relationships from a large time-oriented clinical database: The RX project. Ph.D. thesis, Stanford Univ., 1982 in Computer Science and Biostatistics.
2. Fries, J.F. Time-oriented patient records and a computer databank. *J. Amer. Med. Assoc*. 222, 1536 (1972).
3. Wiederhold, G., and Fries, J.F. Structured organization of clinical data bases. AFIPS conference Proceedings, AFIPS, 1975, pp. 479-485.
4. Blum, R.L. Displaying clinical data from a time-oriented database. *Comput. Biol. Med*. 11 (4) (1981).
5. Teitelman, "INTERLISP Reference Manual." Xerox Palo Alto Research Corp., Palo Alto, Calif., 1978.
6. Kaplan, R. M., Sheil, B.A., and Smith, E.R. "The Interactive Data-analysis Language Reference Manual." Xerox Palo Research Corp., Palo Alto, Calif., 1978.

7. Blum, R.L., and Wiederhold, G. Inferring knowledge from clinical data banks utilizing techniques from artificial intelligence. Proceedings, 2nd Annual Symposium on Computer Applications in Medical Care, Washington, D.C., IEEE, November, 1978, pp. 303-307.

8. Byar, D.P. Why data bases should not replace randomized clinical trials. Biometrics 36, 337-342 (1980)

9. Dambrosia, J.M. and Ellenberg, J.H. Statistical considerations for a medical data base. Biometrics 36, 323-332 (1980).

10. Isselbacher, K.J., el al., "Harrison's Principles of Internal Medicine." McGraw-Hill, New York, 1980.

11. Wiederhold, G. "Database Design." McGraw-Hill, New York, 1977.

12. Stefik, M.J. An examination of a frame-structured representation system. Proceedings, Sixth International Joint Conference on Artificial Intelligence, IJCAI, 1979, pp. 845-852.

13. Kenny, D. "Correlation and Causality." Wiley, New York, 1979.

14. Suppes, P. "A Probabilistic Theory of Causality." North-Holland, Amsterdam, 1970.

15. Mood, A.M., Graybill, F.A. and Boes, D.C. "Introduction to the Theory of Statistics." McGraw-Hill, New York, 1974.

16. Davis, R.B., Buchanan, B.G., and Shortliffe, E.H. Production rules as a representation for a knowledge-based consultation program. Artificial Intelligence 8, 15-45 (1977).

17. Shortliffe, E.H., Davis, R., Axline, S., Buchanan, B., Green, C., and Cohen, S. Computer-based consultation in clinical therapeutics: Explanation and rule acquisition capabilities of the MYCIN system. Comput. Biomed. Res. 8, 303-320 (1975).

18. Armitage, P. "Statistical Methods in Medical Research." Blackwell, Oxford, 1971.

19. Brown, B.W., and Hollander, M. "Statistics: A Biomedical Introduction." Wiley, New York, 1977.

20. Nie, N.H., Hull, C.H., Jenkins, J.G., Steinbrenner, K. and Bent, D.H. "SPSS: Statistical Package for the Social Sciences," McGraw-Hill, New York, 1975.

21. Draper, N.R., and Smith H. "Applied Regression Analysis," Wiley, New York, 1966.

CODA: A CONCEPT ORGANIZATION AND DEVELOPMENT AID

FOR THE RESEARCH ENVIRONMENT

James A. Dewar and James J. Gillogly

The Rand Corporation

Santa Monica, California

CODA: A CONCEPT ORGANIZATION AND DEVELOPMENT AID FOR THE RESEARCH ENVIRONMENT

James A. Dewar and James J. Gillogly

The Rand Corporation

Introduction

In 1982, the authors and colleague Morlie Graubard were led by their experiences to wonder what the role of computers might be in policy research (our primary occupations). The role of computers in analysis, particularly quantitative analysis, is extensive and well documented. Computerized text editing (as used in the preparation of this manuscript) is quickly replacing the typewriter throughout the industrialized world. Artificial Intelligence researchers at Rand and elsewhere are exploring the possibility that computers might do some of a researcher's thinking. Data retrieval systems are bringing vast stores of data within the reach of a researcher's specialized interests. But it seemed as though there ought to be other ways that the power of computers could help the research process.

We focused on the part of the typical research project that involves reading vast amounts of information related to the research topic and remembering the pieces that are pertinent to one's current thesis. This part of research predates computers and is currently supported, to some extent, by a variety of file management and data base management systems. The idea of the computer acting as a long term memory for the researcher seemed like a natural combination of the speed and memory of the computer with the more subtle synthetic powers of the researcher.

In looking at several of the extant storage and retrieval systems, however, they seemed to be directed either toward very large data bases or at personal computer applications. As such, each seemed to have important limitations when applied specifically to the problems of the individual researcher. This led us back to a look at the research process itself in an attempt to define those aspects of search and retrieval that most suited the policy researcher's environment.

The key seemed to be that the research process of an individual was essentially an iterative process characterized by both a growing data base and a series of failed attempts at organizing the data into a coherent whole (followed ultimately by a successful attempt, of course). This led us to an hypothesis about the utility of computers "optimized" for the policy research process and from there to a list of desired characteristics for the associated computer aid.

HYPOTHESIS: Computers can aid the policy research process by acting as a long term memory (storage and retrieval facility) for the researcher's growing data base and changing concepts.

The realization of this hypothesis in the form of computer software specifications required constant referral back to the research process and an appreciation of the limitations of modern computers. The following list of desiderata reflects the results of that effort along with our justifications for each.

DESIDERATA

1. Quick Boolean tag searches.

   In retrieving data, we have found with other retrieval systems that even a 1 to 2 second delay between the request and the results was very distracting to a researcher's thought processes. Boolean searches involve the use of the logical connectives AND, OR, and NOT; require longer than single search requests, in general; yet should respond just as quickly as single search requests if the researcher is not to be distracted from the problem at hand. It should be emphasized that the requirements here are placed on TAG searches — full-text searches are exempted from this specification because it was our thought that the primary search method would be by tags and that the researcher would not mind delays for the occasional full-text search.

2. Flexible tagging rules.

   The process whereby a human retrieves data from his/her own memory is ill understood, except that it seems to involve a variety of mental processes. Retrieving data from a computer is more limited, but we wanted the researcher to be able to "mark" or tag data for retrieval as flexibly as possible. S/he shouldn't be restricted to words that appear in the data or single word tags or . . . .

3. Powerful tag changing capability.

   This would seem to be the key for a policy researcher. As one's concepts change, the relevance of one's data to the new concept usually changes also. We wanted the researcher to have the ability to make wholesale changes to the markers or tags on data. This seemed to imply two capabilities: 1) The ability to do both full-text and tag searches and then to "prune" or shape the resulting list of data records, and 2) the ability then to make changes on the tags of those records all at once.

4. Recall by date capability.

   This is a common capability in storage and retrieval systems and seemed like a useful adjunct for a research environment that depends heavily on dated articles. This also includes the ability to do comparison operations on dates (such as, all dates from date 1 to date 2, etc.).

5. Data entry from keyboard or file.

   This would seem to be useful for two reasons: 1) Many researchers already have electronic data bases that would benefit from this capability, and 2) Rand (among others) now has the capability to get the results of large data base searches in electronic form, which could then be easily entered in this form.

These, then, formed the basic capabilities that we thought defined a computer capability "optimized" for the needs of the policy researcher. With these specific capabilities in mind, we again looked at the available software for file and data base management. Without claiming we did an exhaustive search, we did talk with a variety of data base specialists, search the on-line literature and visit computer shows. In the end, we were sufficiently disappointed with the matches between our desiderata and the available software that we decided to build our own. As a point of interest, the common mismatches were either slow response times (typical of microprocessor based systems) or insufficient capability to do easy, wholesale changes to the tags in the system (typical of systems aimed at very large data bases).

The resulting system was called CODA (for Concept Organization and Development Aid) and that system is the topic of this paper. In the following sections we will describe the prototype system we built for testing our hypothesis, the system's capabilities and limitations, some of the details of its user interface, what we have learned both from the building and testing of the system, and, finally, some thoughts on further capabilities that appear amenable to computer implementation and that might aid the policy researcher.

The Prototype

The CODA program most properly qualifies as a file management system aimed at small data bases and a very limited number of users. The specifications followed in its development were the desiderata listed above and the reason for developing it was to test the hypothesis that the specific task of policy research could be improved by an appropriate computer aid. It is a system designed and implemented by users (policy researchers) for testing some concepts about the users' environment. As such, there are some specific things that CODA is NOT. It is not a full data base management system for general use, it is not particularly suited for large data bases or numerical processing, and it is not a product that the Rand Corporation is trying to sell.

With those caveats in mind, we were interested in describing CODA to the Data Base Management Conference both as a way of highlighting the data base needs on one special segment of the user community, and as a way of eliciting feedback from the professional community of data management specialists.

Before proceeding with the description of CODA, it is important to discuss briefly the major "buzz words" that we have come to use in our description of the system.

tag: This is a user-supplied word or phrase that is typically used in CODA for retrieving a piece of data. In other systems this is called a keyword. 'Tag' is used in CODA because it need not be something that actually appears in a record. A given record can have many tags.

record: This is another word for datum and refers to an individual recallable piece of data in the system. In CODA a record is any (formatted or unformatted) text that is of interest to the researcher. This is intended to include anything from a single word to several paragraphs to a table of numbers to a collection of symbols, etc.

hit: This is any record that contains the specified search tag (or tag combination). If, for example, CODA was commanded to return all records having 'important' as a tag, CODA would return that it had found (say) 34 'hits' or records that contain 'important' as a tag.

index: This is undoubtedly the ugliest of CODA's buzz words. There are two kinds of indices in CODA: date indices and others. Date indices are a way of grouping different kinds of dates for recall. In this way, the user is able to differentiate between, for example, the date that the material in a record was published and the date on which it was entered into the system. The idea of the other indices is much like that of the indices in a book. It is a way of grouping tags both for display and for search purposes. For display purposes, the intention was to give the user something akin to the Author index found in some books. It is much easier to look up a half-remembered author in a smaller author index than it is to do so in the full index. In addition, one can retrieve specifically an author's works, for example, rather than retrieving his works as well as anything written about him.

Technically, the prototype CODA system has been written under UNIX (4.1 BSD) and is running on a VAX 11/780. It is written in C and uses the Curses screen management package. Data entry from the keyboard uses the Rand editor, a full-screen editor used by most of Rand's researchers and secretaries. CODA provides a menu-driven interface to users with a variety of terminal types, including Rand's standard Ann Arbor terminals (in several models) and personal computers connected to the VAX via modems. By servicing all of Rand's terminals, we were able to enlist a variety of Rand researchers to use CODA and feed back their comments as to its utility.

To achieve the recall speed specified in the desiderata, the tags are loaded into a hash table in memory with linked lists pointing to the data associated with each tag. The tags themselves are C strings, which allows for a wide variety of things the system recognizes as legal tags.

Capabilities and Limitations

CODA is a menu-driven system because of our feeling that the typical non-computer oriented researcher would find such a system the quickest and easiest to learn and the most comfortable to work with. The capabilities and limitations of CODA are best referenced directly to those menus that constitute the user interface. Below are the nine CODA menus arranged in a hierarchy based roughly on the connections between the menus.

```
MAIN MENU
        DATA ENTRY Options
        DATA RECALL Options
                HIT LIST Options
                        RECORD Options
                        CHANGE HIT TAG Options
                        REFINE HIT LIST Options
        TAG CHANGE Options
        TAG INDEX CHANGE Options
```

Figure 1. Basic CODA Menu Structure

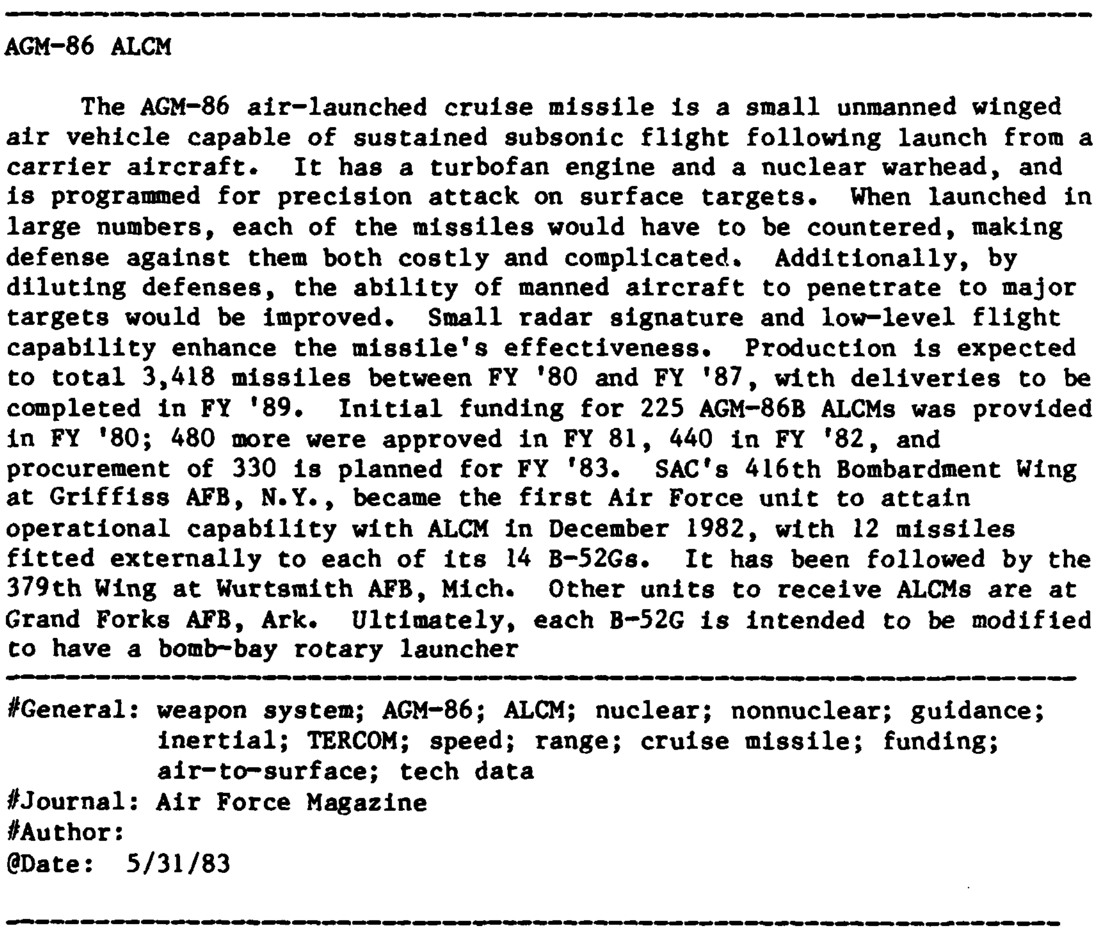
Capabilites

In each menu, typing '?' will give the user a help file that describes the options in that menu, and typing '<ESC>' will get the user out of CODA entirely. The MAIN MENU has an introduction to CODA for the first time user, and is the main access path to the four menus in the first indentation in Figure 1. Of those four, two (TAG CHANGE and TAG INDEX CHANGE Options) relate to tag and index changes throughout the entire data base. The other two (DATA ENTRY and DATA RECALL) are the "guts" of the system, and, along with their submenus, they deserve more detailed mention in order to give the reader a good feel for what CODA is and is not.

```
------------------------------------------------------------------------
                         **DATA ENTRY Options**
  a.  Set up session tags
  b.  Enter data (from keyboard)
  c.  Transfer data (from file)
  d.  Back  to MAIN MENU (*)
Option?
```

Figure 2. Data Entry Menu

Figure 2 is the Data Entry menu exactly as it appears in the bottom window of the user's CRT. Data entry from the keyboard is done in a full-screen two-window editor. An example of a record and its tags (as they have been entered in the two-window editor) is shown in Figure 3.



AGM-86 ALCM

The AGM-86 air-launched cruise missile is a small unmanned winged air vehicle capable of sustained subsonic flight following launch from a carrier aircraft. It has a turbofan engine and a nuclear warhead, and is programmed for precision attack on surface targets. When launched in large numbers, each of the missiles would have to be countered, making defense against them both costly and complicated. Additionally, by diluting defenses, the ability of manned aircraft to penetrate to major targets would be improved. Small radar signature and low-level flight capability enhance the missile's effectiveness. Production is expected to total 3,418 missiles between FY '80 and FY '87, with deliveries to be completed in FY '89. Initial funding for 225 AGM-86B ALCMs was provided in FY '80; 480 more were approved in FY 81, 440 in FY '82, and procurement of 330 is planned for FY '83. SAC's 416th Bombardment Wing at Griffiss AFB, N.Y., became the first Air Force unit to attain operational capability with ALCM in December 1982, with 12 missiles fitted externally to each of its 14 B-52Gs. It has been followed by the 379th Wing at Wurtsmith AFB, Mich. Other units to receive ALCMs are at Grand Forks AFB, Ark. Ultimately, each B-52G is intended to be modified to have a bomb-bay rotary launcher

```
#General: weapon system; AGM-86; ALCM; nuclear; nonnuclear; guidance;
          inertial; TERCOM; speed; range; cruise missile; funding;
          air-to-surface; tech data
#Journal: Air Force Magazine
#Author:
@Date:  5/31/83
```

Figure 3. Date Entry example in a two-window editor

The first line of each record is printed out on the hit list in data recall, so it is commonly used as a summary line for the record. In the tag window, the indices are identified by '#' or '@' and ':' and the tags are separated by semi-colons. These are the only reserved symbols in CODA.

Session tags (option a. in Figure 2.) give the user the ability to set up tags to be put on all data items that are entered until the session tags are changed. This saves the user time in cases where several items to be entered will have tags in common. After setting them up, the tags will automatically appear in the tags window in succeeding data entry calls. Session tags are also useful in data entry from non-CODA files. In order to enter data into CODA from a file, the file must have records separated by a single delimiter that doesn't appear elsewhere in the data. In that case, the records will be entered into the data base with the session tags set up for that purpose.

```
                      ** DATA RECALL Options **
  a.  Enter tags for search
  b.  Look at all system indices
  c.  Look at all tags (for an index)
  d.  Back to MAIN MENU (*)
Option?
```

Figure 4. Data Recall Menu

The Data Recall menu, shown in Figure 4 as it appears in the bottom window of the user's screen, leads into the hypothesized heart of the system — data retrieval and manipulation. In any menu involving tags, the user has the options to look at the current system indices and a glossary of the current tags. Searches can be done by tag, by full-text search, or by Boolean (AND, OR, and NOT) combinations of the two. After the search expression has been entered CODA looks up the pieces of the expression in the tag hash table and offers to do a full-text search if it can't find them. It also will do a pure full-text search. What CODA responds with is shown by example in Figure 5 (in this case the special expression 'all' was entered and CODA returned all the records).

```
There are 278 hits for this expression.
Expression:  all
-----------------------------------------------------------------
   1.  LGM-30F/G MINUTEMAN
   2.  MGM-118A PACEKEEPER (MX)
   3.  AGM-69 SRAM
   4.  AGM-86 ALCM
   5.  BGM-109G GLCM
   6.  AGM-45A SHRIKE
   7.  AGM-65 MAVERICK
   8.  AGM-78 STANDARD ARM
   9.  AGM-88A HARM
  10.  GBU-15
  11.  AGM-109H TOMAHAWK (MRASM)
  12.  ALMV
  13.  AIM-120A (AMRAAM)
  14.  GBU-15s DESTROY SIMULATED MISSILE SITES
  15.  CRUSE MISSILE:  WONDER WEAPON OR DUD?
  16.  BUSINESS OUTLOOK:  CRUISE MISSILE PROGRAM
  17.  INGLORIOUS FAILURES PLAGUE PERSHING-2
  18.  MaRV:  KEEPING A NEW NUCLEAR GENIE IN THE BOTTLE
-----------------------------------------------------------------
                     *** HIT LIST Options ***
  a.  Look at specific record
  b.  Output specific record(s)
  c.  Delete specific record(s) from hit list
  d.  Expunge specific record(s)
  e.  Change tags on these hits
  f.  Refine this hit list
  g.  Back to DATA RECALL (***)
  h.  Back to MAIN MENU (*)
Option?
```

Figure 5. CODA Hit List Example

In some sense, the posited utility of CODA should be measured by the percentage of time the user spends in and around the Hit List menu. It is here that one's ability to rearrange and shape one's "long term memory" is most evident. It is here that we thought the researcher would spend time accessing, modifying and re-marking data in light of new information or insights. Three of the Hit List processing options (a, e, and f) lead to separate menus. The others lead to further prompts from CODA and will be described first.

The records in any subset of the hit list records can be output (option b) in their entirety or to the first blank line (allowing for summary outputs), to a file or directly to the printer, and with or without their associated tags. Any subset of the hit list can be deleted from the list (option c) as a way of shaping the list for output or tagging, or can be expunged from the data base entirely (option d). Paging through the hit list can be done with the +page and -page keys to be found on most terminal keyboards.

Any record can be seen in its entirety (option a) by entering its numerical identifier. This leads to a separate menu (Figure 6). In addition to mimicing the output, delete and expunge options of the Hit List menu, the user can edit the record and its tags (as in data entry), page through the record (if it is greater than one screen-full) with the +page and -page keys, or page through the full records on the hit list (using options b and c). Option g gets the user back to the hit list and its options.

```
                    **** RECORD Options ****
  a.  Edit this record (and tags.)
  b.  Go forward to next hit list record
  c.  Go back to previous hit list record
  d.  Output this record
  e.  Delete this record from hit list
  f.  Expunge this record
  g.  Back to HIT LIST Options (***)
  h.  Back to DATA RECALL Options (**)
Option?
```

Figure 6. Record options menu

Option e on the Hit List menu (Figure 5) gives the user the ability to make tag changes on all records in the hit list simultaneously. This leads to the menu shown in Figure 7. Adding, deleting, or renaming a tag takes place online and is reflected thereafter in any functions that involve the tags.

```
---------------------------------------------------------------
                  **** CHANGE HIT TAG Options ***
  a.  Add a tag to these hits
  b.  Delete a tag from these hits
  c.  Rename a tag on these hits
  d.  Look at all system indices
  e.  Look at all tags (for an index)
  f.  Back to HIT LIST Options (***)
Option?
```

Figure 7. Hit Tag Change Menu

In addition to shaping the hit list by deleting individual hits, option f in the Hit List menu (Figure 5) gives the user the opportunity to refine the hit list using the Boolean operators AND, OR and NOT. This leads to the menu shown in Figure 8.

```
---------------------------------------------------------------
                 **** REFINE HIT LIST Options ****
  a.  Current hits OR those with ...
  b.  Current hits BUT ONLY those with ...
  c.  Current hits BUT NOT those with ...
  d.  Look at all system indices
  e.  Look at all tags (for an index)
  f.  Back to HIT LIST Options (***)
Option?
```

Figure 8. Hit List Refinement Menu

One of the major purposes of options a-c is to allow users unfamiliar with Boolean operations to use them, nonetheless, by having them translated into "natural" English. Option a corresponds to the Boolean

OR, option b to AND, and option c to AND NOT. With these three, the user familiar with logic can build up any desired Boolean refinement and the novice should (with successive refinements if necessary) be able to do the same thing. With each option, the user enters a search expression (as in data recall). CODA then does the appropriate operation and gives the user back the refined hit list, the refined search expression and the Hit List menu.

While CODA has other capabilities, the ones listed above encompass the major ones and those most relevant to the hypothesis that we hoped to test with the system. In addition to the capabilities mentioned, there are limitations worth mentioning also. Some are a direct result of the design chosen to meet the desiderata and others are more subtle in their origin.

Limitations

Perhaps the most se: ious philosophical limitation on the system is that, while it is "optimized" for an individual policy researcher, its assets become increasing liabilities as the number of researchers using the same data base increases. This phenomenon is well known to large data base systems with a large number of users. The more users there are, the more important it becomes to limit both the number of people that can make changes to the system retrieval parameters as well as the frequency with which any changes can be made. CODA is specifically directed at, and can be used profitable by, only a very small number of users PER DATA BASE.

Although not necessarily a limitation, it is true that CODA is not a very "smart" system. Again, it was the intent of the design to leave the "smarts" in this man-machine system in the "man" since that is what "he" does best. The machine has been directed to do what it does very well — store and retrieve data and make changes to the data on command.

The most noticable limitation to the user is that the system can take a long time to load. As currently implemented, the hash table is built each time CODA is operated, hence, the larger the data base and the heavier the machine processing load, the longer the system takes to load. With a 2000 record data base and several tags for each record, the loading can take several minutes on a heavily loaded machine. This is a "start-up" cost and is mainly due to the time required to link the data to their respective tags. While this could be done more efficiently, doing it in C will not be as efficient as it could be in a list processing language such as LISP.

In addition to loading torpidity, the hash table implementation leads to fairly large RAM requirements and to some inefficiencies in truncated searches. While a virtual machine isn't too sensitive to the RAM requirements of any program, the current CODA system set to handle 10,000 records and a hash table set to handle 10,000 tags takes up about 700K in the VAX 11/780. Again, some storage requirements could be gained and efficiency sacrificed by putting the hash table on disk.

Another limitation from the hash table approach is that truncated searches (which aren't currently coded) can't be efficiently coded. An example of a truncated search is to look for all records that contain any word starting with 'bomb'. This would get records with the words bomb, bomber, bombing, bombardment, bombast, bombazine, etc. In full-text search mode CODA can do this quite well, but very slowly. In tag searches, however, it requires searching through the entire hash table, which defeats the purpose of the hash table approach and takes decidedly longer than a single tag search. This is really only a limitation on the tag searches, then, and tends to be a problem with most other approaches that are geared for speed.

The above capabilities and limitations are basically the ones that were designed into the system. They are basically the things that can be said about the system AS SOFTWARE. They are important to reflect upon in terms of trying to understand the tool that was developed to test our original hypothesis. In order to find out if this tool called CODA was of significant use to policy researchers, however, we went to the researchers, had them use it and asked them to give us their opinions on it. In the following section we tell:

What We've Learned

First, it is important to understand that CODA has basically been a "hobby" done on a shoestring budget. These constraints show up in the CODA code as a paucity of "gorilla proofing" and shortage of pre-release testing. They show up in the testing as the lack of a rigorous hypothesis testing methodology and a small number of "beta test sites".

What we have learned about the system comes from the generous assistance of seven Rand researchers (including one of the authors [Dewar] -- from whose work the examples in this paper have been drawn) and a total of nine data bases of various types. As examples, one data base contained data on long range nonnuclear weapons (about 300 records), another contained interview data from Russian emigres (about 2400 records), another contained data on terrorist incidents world-wide (about 1800 records), still another contained information on reviews for the Rand Journal of

Economics (about 400 records), and another contained information on current data bases in the Rand data base library (about 100 records).

Perhaps the most interesting finding for us concerned the size of the data records. The system seemed most useful on records that were small. The two largest data bases had existed previously and were already organized along the lines of "bite-sized" data records and a small number of tags. In CODA the number of tags on these data bases grew slowly as the data was retrieved, sorted and tagged in different ways, but there was no movement toward combining records. In contrast, in one of the smaller data bases that grew with time, there was also a slow move toward splitting large records into smaller, reasonably disjoint records or compacting large records by summarizing them. Records that were on the order of a screen-full or less in size were easiest to scan and absorb quickly.

CODA, as implemented, does indeed appear to be cumbersome for large data bases (which in this case should be taken to mean 1000 records or more). Loading time for the two largest data bases on a loaded VAX 11/780 could be several minutes; looking at the glossary of tags was cumbersome (particularly on the terrorist data where practically every incident had its own unique date identifier); and hit lists tended to be long and cumbersome to wade through while sitting at the terminal. Somewhat surprisingly, however, the users with large data bases have been happiest with CODA. Our guess is that this says much more about the utility of data base and file management systems in general than it does about CODA in particular.

The main "choke point" in the CODA process is definitely data entry and tagging. The ability to enter data from files was the only thing that made testing with the largest data bases possible, but it is a slow and painful process in general to put data into the system and to put tags on it. In somewhat of a surprise, it appeared to be more satisfactory to have a secretary put several records into the system at a time (with a tag such as "untagged") and then to put tags on them by doing full-text searches (on the entire system), than it was to tag them one record at a time.

Originally, non-data indices were a way of grouping tags FOR DISPLAY PURPOSES ONLY. Their intent was to function in much the same way that the Author index in the back of a book does. We found this to be of questionable use and found increasingly that it would be nicer to have the capability to specify at times not only the tag, but also the index with which it was associated. That capability is now implemented, and, in our ongoing tests, is being evaluated.

Finally, in the way of what we have learned specifically, we arrived at some very unscientific estimates of the point at which the CODA system stops being a corroboration of one's own memory and starts to function as a long term memory for things that have been forgotten. There seem to be two kinds of thresholds — one in terms of the number of data records in the system and the other in terms of the time that has elapsed since the first record was entered into the system. In our informal survey it took something on the order of one or two hundred records in the system before the first records has receded enough in a researcher's memory that they appeared fresh when recalled. For data bases smaller than that, it took on the order of a few months of elapsed time before the computer's memory was clearly superior to the researcher's. These estimates claim little scientism, but they do point to the "delay" a researcher can expect before a system like this could be expected to begin paying noticeable dividends.

In using the prototype of the CODA system and getting feedback from others doing the same thing with different kinds of data bases, one not only learns things about the system as it is, but one also starts to get a feel for the general class of improvements that would enhance or establish its utility. It is to this set of reasonable (and not so reasonable) future system possibilities that we now turn.

Wish List

There were three general areas of increased capability that occurred to us during the testing, and in decreasing order of reasonability they can be identified as 1) a bibliographic formatting capability, 2) thesaurus capability, and 3) optical data entry.

In at least two of the data bases there were a large number of direct quotes. While it is possible to put enough tags on these records to enable one to construct a bibliographic reference, there are computer programs available that will construct the appropriately formatted bibliographic reference for you. Data entry on these systems is in "fill-in-the-template" form, and CODA, has the ability to set up user defined templates. While significant work would be required to build a bibliographic formatting capability for CODA, the effort would appear to be worthwhile in future systems of this type.

The second area of improvement that seemed reasonable from our test experience was to build some type of thesaurus capability into CODA. The desirability of a thesaurus that could give "synonyms" for a given tag seems to increase as the size of the tag glossary increases. This would be a slight departure from the philosophical notion that it is best to rely on the researcher for the intelligence in this kind of man-machine system. Nonetheless, giving a CODA-like system some capability to remember or to look for similarities among tags might be a useful "advisory" capability. The details of such a thesaurus capability must remain vague at this point, but some notions of such a capability can be described. One possibility would be to "wire in" a thesaurus, in which case the researcher would be responsible for creating and maintaining the thesaurus and CODA would only respond with synonyms upon request. Another possibility would be to "teach" CODA concepts of similarity and have it constantly review the tags glossary and, upon request, suggest synonyms to the user.

The most desirable improvement comes directly from numerous confrontations with the data entry bottleneck. The most obvious data entry mechanism would be an optical character reader about the size of a light pen that one could use much the way one uses a highlighting marker to mark passages in a text for recall. Entering them directly into CODA in this manner would be a much more satisfactory answer than current data entry methods. While such a capability is an easier technological problem than the even more obvious voice entry capability, the feasibility of such a hand-held mechanism is still, sadly, beyond the current state-of-the-art. In fact, ANY better mechanism for entering data would measurable improve the utility of systems like CODA.

But back to reality. There are some things to be said about the utility of CODA as it is currently constituted and they are the topic of the final section of this manuscript.

Conclusions

Our original ponderings about computer-aided policy research led us down a somewhat tortuous path to the development of CODA. The prototype system was designed to test our hypothesis that computer-aided policy research could be improved and to determine whether our specific set of desiderata for such a system was a path to such improvement. CODA was built roughly to specifications implied by our desiderata, we enlisted Rand researchers to use and comment on it, and we have learned thereby.

As to whether a CODA-like system serves a useful purpose in the community of data base users, our work only leads us to suggestive conclusions. The eagerness of our volunteers to use CODA for more traditional file management purposes and for larger-than-we-envisioned data bases suggests the ever-growing recognition of the utility of computer-aided data management. This, in turn, suggests that specialized communities such as the policy research community are still awakening to the possibilities of computer-aided data management in a variety of forms.

Among the researchers who used CODA as we intended, there was a growing appreciation for and dependence on CODA-like capabilities. This aspect of a growing appreciation tracks with the earlier mentioned note that there is an inevitable time lag between the beginnings of a CODA data base and the appreciation of its long term memory utility.

Several of the desiderata built into CODA appeared indeed to have recognizable utility in the policy research world. The most controversial of these was the on-line ability to change tags simultaneously on large subsets of data. As mentioned earlier, in the general data management world this capability has serious drawbacks. In the community of data bases that have a small number of users, however, this becomes a very powerful tool for reforging the long term memory to conform to the current concerns and theses of the users.

While it is a very subjective judgement at best, search retrievals under one second are a definite improvement over systems with turnaround times only slightly longer. This appears to be much akin to satellite telephone conversations in which a one second delay in conversational responses is distractingly noticeable.

The ability to enter data into CODA electronically from a file was very useful in transferring extant data bases into the CODA system. In addition, this ability led to some serious musings on the integration of CODA-like capabilities with larger data base management systems that have on-line capability for retrieval from very large data bases.

By way of improving the policy research process, the one currently feasible desideratum the CODA prototype seemed to lack was a bibliographic formatting capability. With the addition of this item to the specifications list, the general design goals of a useful policy research computer aid would seem to be complete.

In summary, our work with CODA leads us to believe that there is room for improvement in the area of computerized data management aids specifically designed for the policy research and related communities. While CODA may not be the optimized realization of that opportunity, the desiderata that led to its creation, along with the addition of a bibliographic formatting capability, form an excellent foundation upon which to build such a system.

SCIENTIFIC INFORMATION = DATA + META-DATA

John L. McCarthy

Lawrence Berkeley Laboratory

University of California

Berkeley, California

SCIENTIFIC INFORMATION = DATA + META-DATA

John L. McCarthy

Lawrence Berkeley Laboratory

University of California

1. Introduction

Scientists deal with various types of data, from experimental results to simulation model parameters. Whether jotted down in a logbook or stored on a computer, such data are of little value without some additional information about their content and organization. We call this additional information "meta-data" because it is "data about data." If both are in computer-readable form, meta-data can be used to help automatically manage and manipulate data. But if they are not systematically organized and integrated with the data they describe, meta-data can become part of the data management problem rather than a key to its solution.

Scientists currently employ a wide range of approaches to data management and meta-data. Many still use ad hoc procedures implemented in FORTRAN and other general purpose programming languages -- in part because they have data structures and usage patterns that cannot be easily accommodated by general purpose database management and statistical analysis systems [Shoshani et al. 1984]. Some in certain disciplines (e.g., health sciences) make extensive use of statistical package programs such as SAS [SAS 1982] and SPSS [SPSS 1983]. A few use data management systems such as INGRES [Relational Technology 1984], RIM [Boeing 1983], RS/1 [BBN 1983], SIR [Fox 1984], and SPIRES [SPIRES 1983].

Although each of these approaches has its respective merits, none of them provide the full range of meta-data facilities that many scientists need. At the same time, few scientists have recognized that many of their data-related problems call for both improved use of existing tools and development of better tools for integrated management of data and meta-data.

Meta-data problems impact both productivity and data quality. In meta-data are not generated as an integral part of the original data collection, processing, and analysis effort, certain types of information may be lost or prohibitively expensive to recapture. If data and meta-data are not closely tied together, it is difficult to track down the origin of questionable results or to estimate the reliability of key findings.

Inadequacies in the form or substance of meta-data also inhibits data sharing. If they are not computer-readable, subsequent users have to spend additional resources putting meta-data necessary for data analysis and display into machine-readable form. If meta-data do not provide full documentation on how data were collected and processed, other scientists cannot assess potential data usefulness and reliability for themselves.

The purpose of this paper is to improve understanding of meta-data in general and scientific meta-data in particular. Its longer term aim is to improve scientific data quality and research productivity through better meta-data, meta-data practices, and software tools for integrated management of data and meta-data. It builds on and complements previous work at Lawrence Berkeley Laboratory (LBL) concerning statistical data [Shoshani 1982], meta-data [McCarthy 1982b], self-describing files

[Merrill & McCarthy 1983], and scientific data characteristics [Shoshani et al. 1984].

In order to provide a basis for more detailed discussion, section 2 introduces key concepts about data and meta-data in terms of several examples. Section 3 outlines a typology of meta-data, with special emphasis on scientific and technical needs. Section 4 describes how these different kinds of meta-data can be used at various stages of scientific research and analysis. Section 5 discusses principles for representation of meta-data. Section 6 considers how self-describing files, data management systems, statistical analysis packages, and data analysis environments can be used to integrate management of data and meta-data. Section 7 presents brief summary conclusions.

2. Data and Meta-data Examples

To provide useful information, computer databases and even handwritten datasets must include two logical components — data and meta-data. Meta-data are descriptions which help people and computer programs understand and manipulate the data they describe. Meta-data may be as rudimentary as handwritten notes describing a set of instrument readings or as structured as database schema definitions in an information management system. The examples below illustrate and elaborate on these points.

2.1 A Simple Example

Like data, meta-data are most useful when they are systematically organized in computer-readable form. For example, the small "self-describing" file in Exhibit 1 contains an initial section of meta-data

Exhibit 1: Self-describing File of Meta-data and Data *(with comments in italics)*

```
                                             FILE LEVEL INFORMATION
file name = aqsols                           name of this file
  description = Aqueous Solutions Database
  contact = Sidney L. Phillips
  phone = (415) 486-6865
  last modified = 11/2/84
  global missing code = -999                 default value for all elements
                                             FIELD LEVEL INFORMATION
 data element = Element Name                 name of first field
   type = char                               data type is alphanumeric
   start = 1                                 field begins in column 1 of record
   length = 11                               maximum length of field is 11 columns
 data element = Atomic Number                second field
   type = int                                data type is integer
   start = 13
   length = 3
 data element = Formula Mass                 third field
   type = float                              data type is floating point
   start = 16
   length = 9
 data element = Electronic Configuration     fourth field
   type = char
   start = 26
   length = 20
   missing = "not available"
 data element = Electronegativity            fifth field
   type = float
   start = 47
   length = 5
ENDDF                                        end of data definition (meta-data)
                                             DATA
Arsenic     33  74.9216  3s2 3p6 3d10 4s2 4p3   2.0
Bromine     35  79.9090  3s2 3p6 3d10 4s2 4p5   2.8
Chlorine    17  35.4530  2s2 2p6 3s2 3p5        3.0
Carbon       6     -999  1s2 2s2 2p2            2.5
Phosphorus  15  30.9738  3s2 3p3                2.1
Silicon     14  28.0855  3s2 3p2                1.8
```

which describes the three rows of numeric data at the bottom of the exhibit. The meta-data specify how the data are formated and what they represent. They provide non-procedural instructions about how to manage and manipulate their associated data for a computer program designed to interpret such information. (Explanatory comments in italics at the right side of the exhibit are not part of the input data or meta-data.)

Some types of meta-data such as data type and length of data fields are necessary primarily for computer programs that read and process the data. The "global missing code" statement specifies particular values of the fields that should be treated as "missing data" in manipulation and analysis routines -- e.g., the third field (Formula Mass) for the fourth (Carbon) data record. Other meta-data such as the file description and contact information are primarily intended to help human beings understand data content and output produced by analysis and manipulation programs.

This non-procedural approach to data definition, in which users specify what they want done, without having to spell out the details of how to do it, typically gives faster results with fewer errors. Users can employ a general purpose system to read and manipulate the data without having to write a more extensive and detailed procedural program to accomplish the same work. Moreover, some systems can operate simultaneously on data and meta-data, so that the output is "self-describing," in the sense of the above example and as discussed further in section 6 below.

2.2 A More Complex Example

Most scientific data is more complex than our initial example. For instance, consider a typical scientific experiment in which different instruments automatically collect various types of data. Supporting information about the instrument set-up itself is kept in a separate text file. Instrument calibration values taken during the course of the experiment reside in yet another file. Anomalies that arise in particular instruments during the course of the experiment are noted in a handwritten log-book. Data are subsequently processed and summarized for analysis by FORTRAN programs and statistical analysis packages, each of which contains additional control information (e.g., thresholds and other validation criteria, conversion factors, etc.) embedded in data statements, constants, input parameters, and auxiliary files. Metadata, although not explicitly identified as such, are scattered through the text of documentation files describing the various data formats and procedures.

In this example there are several related types of data and metadata. We can categorize them using a modified version of the typology suggested by [Shoshani et al. 1984]. First, there are the basic experimental data collected by the instruments. Second, there are associated data, including descriptions of the data collection instruments, their configuration, initial settings, periodic calibrations, and notes on anomalies that arose during the course of the experiment. Third, there are analytic control data used to reduce, edit, and analyze the raw experimental results. Fourth, there are intermediate summary data produced from the experimental results (e.g., data on individual particle tracks inferred from numerous "events" recorded by individual detectors). Finally, there are meta-data in the documentation files which describe the data files, fields, and their inter-relationships.

This example also illustrates a number of potential meta-data problems. The different types of data and meta-data that describe them are not stored in a consistent manner (e.g., handwritten notes, unstructured text files, numeric data files, program statements, etc.). Some important meta-data are altogether lacking; there is no easy means of identifying the inter-relationships between different types of data (e.g., common key fields). Without an integrated approach to data and meta-data, each processing step will be difficult, time-consuming, and prone to error. Each additional step is likely to compound the problems. The more complex the data, the more urgent is the need for meta-data to tie it together and software to manage both in a coherent way.

3. Major Types of Scientific Meta-data

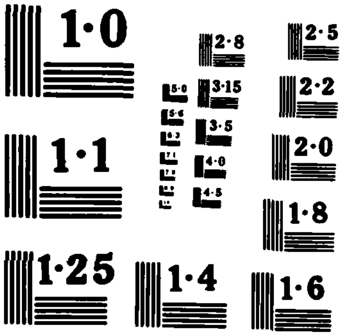
People and computer programs need different kinds of meta-data in order to understand and process data. At present, most data management systems support few types of meta-data beyond those essential for standard computer processing (e.g., type, length). Statistical analysis systems typically add some meta-data for statistical computing (e.g., missing data codes, variable value labels). Various data dictionary and specialized application systems support additional types of meta-data, but still with a primary emphasis on data processing requirements.

People need a wider variety of meta-data to answer the classic questions "who, what, how, where, and when" in searching for and using data. Toward this end, Dolby has suggested that each datum, table, and dataset should be associated with specific meta-data to provide classification by source, observer, matter, function, space, and time [Dolby 1984]. Many such types of meta-data turn out to be useful for computer programs as well. For example, if meta-data about a particular variable include whether it is nominal, ordinal, or ratio level, analysis programs can use that information to warn the user if improper procedures are invoked (e.g., taking means of nominal variables). More detailed consideration of overlapping meta-data uses appears in section 4 below.

Some types of meta-data (e.g., names and labels) may apply to an entire file or database, to a particular data element or field, or even to a single datum (in which case it is usually part of the data). In some cases distinctions between data, associated data, and meta-data are fuzzy at best. A particular item may be data for one purpose and meta-data for another. These issues and ways in which higher (e.g., file) level information can be specified to apply to lower levels (e.g., fields) through "inheritance" or default values are discussed in more detail below in section 5, along with other representation issues.

There are many ways to classify different types of meta-data. This section attempts to do so in terms of function, with an emphasis on those most relevant for scientific and technical data. It is by no means exhaustive. The intention here is simply to outline the major kinds of information that scientific meta-data may include. Where possible, it includes references to more detailed work as well as to specific systems that support particular types of meta-data.

AD-A158 285 DATA BASE MANAGEMENT: PROCEEDINGS OF A CONFERENCE 2/4
NOVEMBER 1-2 1984 HELD AT MONTEREY CALIFORNIA(U)
STANFORD UNIV CA DEPT OF STATISTICS H SOLOMON ET AL.
UNCLASSIFIED 31 JUL 85 N00014-76-C-0475 F/G 9/2 NL


1·0
2·8
2·5
3·15
2·2
3·5
2·0
4·0
1·1
4·5
1·8
1·25
1·4
1·6

Value Label (Category) Sets

Several statistical systems permit users to define sets of labels corresponding to sets of numeric or coded values of specified variables. These value label sets automatically provide row and column identifiers for cross-tabulation, analysis of variance, and similar operations [SPSS 1983, Buhler 1979, Institute for Social Research 1973, Becker and Chambers 1984]. Use of descriptive information vectors can also be extended to label and describe subscripts of data elements or fields that are themselves vectors or multi-dimensional arrays [McCarthy 1982a). For example, two category sets -- RACE (white, black, other) and AGE (99) -- might be used to define, describe, and label a two dimensional array (3 by 99) of mortality rates. In certain cases, such information can also be used to aid automatic data conversion between different category sets (e.g., from one set of age groups to another) or from one logical data structure to another.

Labeling and Descriptive Information

Other descriptive meta-data items that can supplement names, labels, and aliases at various levels include a one-line title (e.g., "Carbon to Nitrogen Ratio"), table stub and column labels, unlimited textual description, subject index terms, special remarks, and footnote references (numbers or letters referring to a common set of footnotes).

3.2. Data Derivation and Quality Indicators

Information about derivation and quality is crucial for scientific data. Meta-data for such purposes will vary according to the needs of individual projects. Some specific examples are as follows:

3.1 Identification, Description, and Labeling

Most current systems support little more than 2 to 40 character names for description of fields, files, and similar entities. Documentation and display of scientific and technical data demands a richer variety of descriptive meta-data, organized in discrete components so it can be used for different purposes. Scientific descriptions often include Greek characters, subscripts and superscripts, and other mathematical notation. Although scientists have found ways to represent necessary notation using the limited character sets of line printers, it is preferable to store, retrieve, and print scientific notation insofar as possible — either directly or via some kind of pre- and post-processing. Some of the most important specific types of descriptive meta-data are as follows:

Names and Aliases

Names are short words or character strings that identify something such as a data field or database (e.g., CN-ratio", "Particle Properties"). In scientific applications, identifiers are frequently made up of hierarchically ordered names or codes (e.g., class/type/sub-type/form = Metal/Steel/Low Alloy/ AISI4340/bar). Aliases or synonyms are alternative identifiers that can be used in place of main names (e.g., "CN"). Ability to add multiple synonyms easily helps adapt names to individual user needs without having to make people agree on using a single term for each thing.

Variable Creation and Modification History

Some special purpose statistical analysis systems include capabilities to document creation of derived variables by automatically keeping the arithmetic and logical operations used to create them (e.g., CN-Ratio = Carbon/Nitrogen) [Stewart 1978, Shanks 1980]. A general capability of this sort might recursively look up any additional derivations involving variables shown on the right hand side of each equation.

For many scientific projects it is also important to keep an audit trail of data modifications. Although a number of systems provide logging to a separate file, information about who made what changes, when, and why may also need to be summarized in the meta-data for each file or variable.

Data Quality and Reliability Indicators

It is sometimes possible to assign a data quality indicator to a particular database, variable, or value. Such indicators can be used for weighting, case selection, and other purposes, [Stewart 1978, Klensin 1981]. For example, one indicator might distinguish three levels of reliability: "best guess," "based on limited experimental results," and "based on many critically evaluated experiments."

Error Function

In some cases (e.g., stratified random sample data), the estimated error of individual data values can be computed from a mathematical formula involving other data elements [Sparr 1982, Klensin 1981].

Instrumentation Information

As mentioned above, it is important to know what kinds of instruments have been used to collect data, how they were calibrated, etc. In some cases this may require an extensive associated database. In others, simply the name of the instrument and a few parameters may suffice.

Bibliographic Citations

Citation meta-data can link data to the sources from which it was drawn, or to related publications.

Personal Names and Contact Information

In some cases meta-data can include contact information (e.g., name, telephone numbers, electronic and regular mailing addresses, etc.) for persons who create, modify, or know about specified aspects of the data.

3.3. Analytic Information

Some types of meta-data are particularly important for people and computer programs that deal with data analysis.

Units of Analysis or Universe

Different fields within a single data record can pertain to different underlying universes or populations within the entity that the record represents. For example, the universes for two sets of fields in some epidemiological data for geographic areas might be "total population," and "Hispanics over 18." Such information is frequently overlooked when it is simply included as part of a title. As a separate meta-data item it can be used for type-checking and indexing.

Measurement Units

Standardized meta-data identifying units of measurement (e.g., gallons, miles, count of persons, etc.), can be used by computational routines to provide automatic conversion (e.g., miles to kilometers), and extended type-checking when performing operations involving different attributes [Sparr 1982].

Missing Data Specifications

Most statistical packages and some data management systems provide support for a valid value range or list of missing data codes. These can be used to select valid cases and to flag conditional processing procedures in certain statistical routines [SPSS 1983, SAS 1982, Buhler 1979, Fox 1984]. Another approach found in some database systems is to specify certain default values or procedures to be invoked when a data value is null or falls into a category otherwise defined as "missing."

Weighting Expression

Another type of meta-data frequently found in statistical packages is a weighting factor to be applied to individual instance values of a particular field, to take into account such factors as scaling of values to save storage space, or disproportionate stratified sampling in survey data [SPSS 1983, SAS 1982, Institute for Social Research 1973, Buhler 1979, Fox 1984]. Weighting expressions can be simple constants (sometimes called scale factors) or a functional expression involving data values from other fields.

Statistical Summaries

For purposes of summary description in very large databases, meta-data may also include univariate statistical summaries (e.g., median, selected percentiles, range, count of non-missing data values, count of unique values), for selected data fields, record subsets, etc. Some have even suggested including multivariate summaries of relationships (e.g., correlation coefficients). If data are not relatively static, this involves non-trivial update considerations [Rowe 1982, Boral 1982].

3.4. Processing and Security Procedures

Scientific data frequently require sophisticated checking and transformation of input, as well as various types of output processing (e.g., to produce scientific notation). Data that are used by more than one user require protection mechanisms as well. For data integrity and documentation, it is useful if specifications about how data are transformed, checked, and protected are included as part on non-procedural meta-data.

Input, Output, Indexing, & Searching Procedures

Processing procedures for individual fields such as simple recoding, transformation, and checking of data on input are found in most data management and statistical packages. A few data management systems such as SPIRES provide more powerful and extensive sets of standard data item manipulation functions for input, output, indexing and query modification [Schroeder et al. 1975, SPIRES 1983]. Standard mechanisms for conversion of different representations of measurement (e.g., time, mass) to and from an internal canonical form (e.g., SI units) are particularly important for scientific data.

Security Specifications

Security specifications designate what users or programs can read, add, delete, or modify designated types and levels of information (e.g., particular fields with values in certain ranges).

Usage Information

Usage information may include which programs access, update, or control which meta-data entities and vice-versa, summary usage statistics such as number of accesses, etc.

3.5. Physical and Structural Characteristics

Most systems include meta-data to describe simple physical characteristics such as field lengths, data types, etc. Few support complex objects, abstract data types, and information about logical structures. Meta-data of importance for scientific information in this area include the following:

Physical Characteristics

In addition to standard physical characteristics such as field type and maximum length, meta-data can specify whether fields are variable or fixed length, and whether they can have more than a single occurrence (e.g., as in the case of synonyms). Disk dataset names, block sizes, compression algorithms, etc., also fall under this heading.

Logical Structure Description

Structural information may include multiple logical views for different users and descriptions of complex data types, such as arrays. For scientific databases, it is important that users be able to access and manipulate entire structural units as objects (e.g., vectors, matrices, and non-homogeneous data structures such as analysis of variance tables) [Becker and Chambers 1984, McCarthy 1979, Klensin 1981].

Access Path and Linkage Specification

In addition to structural characteristics of individual databases and attributes, meta-data can also contain descriptions of how different entities are linked, (e.g., which attributes determine the unique key of a particular record-type, or what access path represents a particular many-to-many relationship).

4. Meta-data Uses

Meta-data constitute a crucial component of scientific information because they facilitate data validation, editing, documentation, manipulation, storage, retrieval, analysis and display. Meta-data can help people and computer programs define, locate, and control data at different stages of scientific inquiry, from experimental design to retrieval of archival information.

4.1. Functional Uses of Meta-data

For scientific as well as other fields, functional uses of meta-data fall into four broad categories:

definition and documentation;

quality assurance and security;

selection and manipulation;

output and display.

In each of these areas, the power and utility of meta-data for increasing information quality and productivity depends primarily upon (a) quantity and quality of meta-data; and (b) computer software that can effectively utilize meta-data information.

Definition and Documentation

Meta-data can describe and document not only internal system data files, but also external data to be loaded or data output for export to other systems.

If well differentiated and structured, meta-data can be used to generate a full range of both printed and on-line documentation of different types, including data dictionaries, indexes, thesauri, etc. One important role of documentation is to provide information about data sources, collection procedures, codification methods, and quality controls so that people can understand the derivation and limitations of data they may wish to use.

Quality Assurance and Security

In addition to passively documenting data sources and processing procedures, meta-data can be used to actively specify and control various quality checks and integrity constraints. Validation constraints can be automatically enforced whenever new data is input. Security-related meta-data can specify which people and programs are permitted to read, add, modify, or delete what data, so that the computer system itself can enforce those constraints. Still other meta-data can be automatically generated to indicate who modified which database components at what time.

Selection and Manipulation

Meta-data play a key role in selection and manipulation of data. They provide a summary overview of data content and organization -- just as tables of contents, indexes, and tabular headings do for printed materials. They can enable users to browse, retrieve, and manipulate data of interest with simple field names, logical operators (e.g., ">" and parentheses), and functions (e.g., "sort"). For example, users may employ meta-data entity and attribute names along with attribute values to select data while browsing an on-line database dictionary or via a direct query, such as

```
find materials where tensile.strength > 60 ksi for temperature -400:-300
```

Display and Output

Scientific data analysts often create tables, graphs, and other types of output which must be labeled and documented. If data, meta-data, and display software are integrated, the display software can automatically generate labels for variables, measurement units, and other ancillary information from meta-data associated with any given subset of data. Meta-data also can provide control information such as default field widths, line characteristics, grid size, etc. for output displays.

4.2. Scientific Uses of Meta-data

In scientific settings, meta-data can play important roles at each stage of research. At present, few scientific projects exploit the full potential of meta-data at each stage. In the future, however, meta-data can play a key role in integrating information over the entire research process.

Hypothesis Formulation and Experimental Design

During initial stages of research, meta-data can help facilitate retrieval of relevant publications, information about prior experiments, and necessary baseline data. Bibliographic meta-data identifying what databases cover which publications, what information each field in a database contains, and so on, facilitates retrieval as well as merging of information from different sources. As critically evaluated numeric databases become available, their meta-data will permit scientists to browse and retrieve up-to-date information formerly located in handbooks and similar sources. For research that requires information about experimental design, meta-data can help scientists locate information from prior experiments in public or private data archives. For example, an investigator might be able to learn what kind of instruments gave the best precision under certain conditions in prior experiments.

Instrumentation and Data Collection

Associated data about experimental design, instrumentation, and data collection methods is very important is scientific research. Sometimes, as in the case of detectors for high energy physics experiments, it can present a significant data management challenge in its own right

[Shosani et al. 1984]. Appropriate meta-data management tools can facilitate recording of instrument locations and calibrations as well as changes that occur during the course of an experiment, along with automatic time-stamping and identification of which person or program noted such changes. As mentioned above and discussed further below, meta-data can contain specifications for quality control and integrity checks as data are collected and subsequently processed. If kept in human-readable form as part of a central data dictionary or similar facility, such information also provides documentation for subsequent research stages.

Analysis and Display

Meta-data have been recognized as an important component of data analysis by many researchers. Part of the popularity of statistical packages is due to their facilities for data specification, labeling, and documentation of analyses, including tabular and graphic displays. As discussed in more detail below (in section 6.5), the new generation of interactive data analysis software environments that scientists are beginning to use put more emphasis on integrated treatment of data and meta-data.

Codification and Archiving

Meta-data are even more essential as data passes from scientists who carried out the original research to others who wish to codify or re-analyze selected subsets of the data — often for purposes other than those of the original research. Archival documentation is considerably easier if meta-data have been systematically assembled and catalogued at prior stages. Archival meta-data helps subsequent users find, retrieve, and merge the data with other results, thus beginning the next cycle of research.

5. Meta-data Representation

One of the chief barriers to integrated use of meta-data has been the difficulty of representing them in standard, computer-readable forms. One the one hand, meta-data needs much the same systematic, standardized treatment as data. Computer representation can help facilitate as well as enforce such standardization. On the other hand, meta-data differ significantly from data in both structure and content. Regardless of what tools are employed (e.g., FORTRAN programs, data dictionaries, data management systems, or statistical packages), better understanding of representation issues can improve how we manage and use meta-data. Representation principles are particularly important for scientific meta-data because scientific applications tend to require more extensive and varied meta-data (as discussed in sections 3 and 4 above) than standard data processing.

What are the primary differencies and similarities between scientific data and meta-data? The majority of scientific data are numbers, with some smaller proportion of codes which may be alphanumeric. Most of them can be represented easily in singly occurring, fixed-width fields. They can be pictured logically as tables, with cases or observations as rows and variables or fields as columns. Meta-data on the other hand, contain much higher proportions of variable length, multiply occurring text fields and structures. It therefore is frequently clearer to picture meta-data in a structured "name = value" form, as in exhibit 2. In this example, note the variable-length description and footnote items, the "name/phone" structure, and several meta-data fields such as "synonym," "footnote," "citation," and "name/phone" that take on multiple values.

Exhibit 2: Extended Meta-data for One Field *(with comments in italics)*

```
data element = elongation
  synonym = elongation at break                 note three different aliases
  synonym = ductility
  synonym = tensile ductility
  type = float
  start = 51
  length = 9
  missing = -998
  units = percent
  quality = evaluated
  description = the increase in gage length of a tensile test specimen,
            usually expressed as percentage of the original gage length
  footnote = the increase in gage length may be determined either at or
          after fracture, as specified for the material under test
  footnote = in reporting values of elongation, the gage length shall be stated
  citation = LBL-14996                          keys to entries in another database file
  citation = ISBN-0-201-14474-3
  contact = John McCarthy                       note repeating name/phone structure
    phone = (415) 486-5307
  contact = Edgar Expert
    phone = (800) 555-1212
```

The primary objects that meta-data describe are data fields (variable, attributes) and files (relations or databases). Subsidiary meta-data objects may include data records (cases or observations), category sets (value labels, etc. as described in Section 3), groupings of fields, citations, and footnotes. Meta-data also describe <u>relationships</u> between these various meta-objects. For example, in exhibit 2, the two "citation" meta-data field entries are keys of records in another database that contains full bibliographic information for each citation.

How can we represent such information in computer readable form so that meta-data can be used to manage and control the data they describe, and so that transformations of data automatically generate corresponding meta-data? The remainder of this section outlines a series of representation principles that extend ideas about data representation (e.g., a particular type of data should appear in the same place and same format in each record) to meta-data. Most of these principles apply equally to data as meta-data, and in fact may be helpful for types of data that are currently considered non-standard (especially textual data). These principles are also independent of type of data or degree of computer automation.

General principles for representation of meta-data can be summarized under three headings: clarity, maintainability, and flexibility. Each of these principles can be broken down and defined in terms of three more specific and detailed requirements, as follows:

5.1. Clarity

Above all, meta-data must be clear for both people and computer programs. In some respects, people have different requirements than programs for optimal representation of meta-data. People prefer readable, structured text. Programs require standardized, structured data. For both, however, clarity implies differentiation, structure, and simplicity.

Differentiation

Meta-data need to be decomposed into distinct, standardized semantic components. In computer language design, this basic principle is usually referred to as "orthogonality." In a recent article on database language design principles [Date 1984], Date suggests that a better term might be "concept independence: Distinct concepts should always be cleanly separated, never bundled together." For example, instead of including measurement units simply as part of a label it should be a separate meta attribute of an individual field or variable (e.g., "label = melting point; units = degrees centigrade" rather than "label = melting point in degrees centigrade"). Including "units" as a separate, optional meta-attribute for each data field not only is clearer for human users, it also facilitates computer processing (e.g., automatic unit conversion).

Meta-data that has been broken down into highly differentiated components can always be recombined. It is much more difficult to extract meta-data components from larger undifferentiated units, let alone to use such information in quality control constraints or other operations.

Unfortunately, most current data management and statistical systems do not provide facilities for highly differentiated meta-data. At best, users have to bundle information into comments and labels. However, some recently developed "data analysis environments" promise to provide more extensive and extensible capabilities (see section 6.6 below).

Structure

It is not enough simply to decompose meta-data into many conceptually distinct components. Human beings have difficulty dealing with more than five to ten items at the same level of detail [Miller 1956]. To preserve clarity along with differentiation, meta-data need to be structured into groups of related information and hierarchies of such groups. For example, exhibit 1 illustrated one common way of organizing information about a data file, with information about the file as a whole together at the top, followed by groups of information for each field.

Meta-data items are easier for humans to comprehend if they are organized in a standard order and format. For example, the meta-data for each field in exhibit 1 always appears in the same order, with the same indentation. Order and indentation are not necessarily important for computer programs, but spelling is more crucial than for human beings. Programs can reformat meta-data into easier-to-read indented formats so long as each piece of information is distinct and correctly identified.

Simplicity

Clarity also requires simplicity. Even though highly differentiated and structured, meta-data also needs to be simple to represent and use. Syntax should avoid use of special symbols, abbreviations, idiosyncratic conventions, rigid fixed-field formats, and the like. One effective type of representation is "name = value" (e.g., "missing = -999" or "label = melting point"), which both human beings and computer programs can interpret relatively easily. Representing meta-data by simple text strings makes them easy to modify and maintain with standard tools that operate on text. It also makes them easy to translate into more efficient forms of representation for computer processing (see section 5.3 below).

5.2. Maintenance Considerations

Meta-data are subject to change, so they must be easy to maintain. For ease of maintenance, meta-data components need to be integrated with each other and the data they describe, authoritative in terms of actively controlling that data, and parsimonious in their representation.

Integration

Meta-data need to be closely linked to each other and to the data they describe so that users can comprehend them as a coherent whole and programs can use them actively in conjunction with data and one another. For example, if there are constraints on what values are valid for a given field (e.g., only "0" or "1"), that needs to be explicitly stated in the meta-data, and such meta-data should be actively used to guarantee that no other values can be entered in the field. Similarly, a report generation program should be able to access column and row labels, measurement units, output field width information, and any relevant footnotes in order to construct tabular data displays.

It should also be possible to link data and meta-data items in a variety of ways. In some cases it is useful to pass values from a number of different meta-data items (e.g., name, subject, description) to a single over-all (word) index. In others it is desirable to pass values from a certain type of meta-data to several different distinct indexes (e.g., "field name" might be passed to a "field name index" as well as to an over-all "word" index). If data and meta-data are stored separate from one another, there need to be linkages so that users and programs can operate on both as a single logical unit (see section 6 below).

Authority

Different representations of the same meta-data information should all be generated from the same authoritative source. Usually one representation is required that human beings can read and edit (e.g., a structured ASCII text version), while another is required for computer efficiency (e.g., a "compiled" version). So long as one representation (i.e., the editable one) is recognized as authoritative, computer software can facilitate and enforce synchronization. Some database systems store both textural and "compiled" versions of meta-data in standard database records so that standard update constraints can guarantee correct synchronization.

The problem becomes fuzzier and more difficult if there are redundant editable versions of the same information. For example, derivation of new variables may be documented in terms of algebraic expressions, although the actual creation of new variables is specified by a computer program written in FORTRAN or a special report generation dialect. This kind of discrepancy between documentation and what it purports to describe is a common (and insidious) meta-data problem. It is similar to what happens when a programmer modifies code produced by an application generator or pre-compiler (e.g., RATFOR); there is no longer a single authoritative source version. Although it is not easy to translate algebraic expressions that non-programmers can understand into computer instructions, meta-data that a user reads as documentation should not differ in substance from what computer programs use for controlling and manipulating data. Ideally, the program should be compiled directly from specifications used for the documentation in order to guarantee consistency.

Parsimony

Parsimonious representation makes it easier for human beings to understand and maintain meta-data. One important aspect of this principle is to minimize redundant information. For example, meta-data that are identical for most fields (e.g., missing data codes) can be specified at the file level but over-ridden for individual fields (as in the "Electronic Configuration" field of exhibit 1). This notion of attributes that to pertain to whole classes of meta-entities, termed generalization by data management researchers [Smith 1978], is closely related to the artificial intelligence concept of "inheritance" from one class to all those below it.

Parsimony also bears on the more general issues of logical representation and meta-data maintenance tools. Given their characteristics of inheritance, multiply-occurring values (e.g., synonyms), and nested structures, it is usually more parsimonious from the user's point of view to represent meta-data as hierarchical or network structures rather than fully normalized relations. In the same spirit of parsimony, users should be able to use the same software tools for entry, indexing, searching, retrieval, updating, and display of meta-data as for data. One way to facilitate these goals is to use the same storage structures and management mechanisms for data, meta-data, and meta-meta-data.

5.3. Flexibility

Representation mechanisms should be sufficiently flexible to deal with diverse current requirements as well as unanticipated new needs. In particular, meta-data representation should facilitate transformation from one form of representation to another, recursive reference of one type of meta-data by another, and user-defined extensions to represent new types of meta-data.

Transformability

For human users, structured text and graphical representations probably are the optimal formats for meta-data. For computer processing, storage, and communications, however, binary tables and lists are more efficient. Meta-data therefore need to be amenable to transformation between different forms of representation. They also require tools (filters) to facilitate such transformations in both directions. As in other situations where there are parallel versions of the same information (e.g., a high level application generator specification, source code produced by the application generator, and a compiled version of the source code), care must be taken to synchronize the different versions.

Users also need to enter and output meta-data in different formats. For example, in some instances it may be easiest to enter meta-data on a field by field basis, while in others it may be easiest to enter all of a given type of meta-data for all fields together. Some users may prefer formats that make more use of special symbols and conventions (e.g., "range = 1:99," rather than "maximum = 99; minimum = 1"). If meta-data are well defined and differentiated, existing software tools (e.g., awk or sed under Unix) can easily perform conversions between different input and output formats.

Recursive Reference and Expressions

Certain kinds of meta-data need to contain "executable" arithmetic expressions and references to other meta-data. For example, virtual fields can be created on the fly if there are facilities to calculate them from expressions such as the following (where "mpct" and "stress" are the names of data fields and "sqrt" is a built-in function):

error = sqrt(4*mpct*(1-mpct)/stress)

As noted in section 3 above, such expressions occur in a number of types of scientific meta-data.

The clearest and simplest way to define meta-data from a user standpoint is to employ a core set of meta-data to define itself as well as other meta-data. Some database systems permit at least a rudimentary capability of this sort. Recursive definition also facilitates user-defined extensions to meta-data and meta-meta-data.

Extensibility

Since we cannot hope to anticipate all the possible types and forms of meta-data, it is essential that representation mechanisms be as extensible as possible. Scientific users in particular need to be able to add new types of meta-data without having to recompile new versions of the entire system or reload existing database every time they do so.

If meta-data and tools to use them are to be extensible, simplicity of representation becomes even more important. The simple "name = value" form facilitates easy scanning, storage, indexing, and retrieval of new types of meta-data. If different types of meta-data require different syntax, each addition to existing meta-data may require changes in parsers, translators, and other software.

6. Co-management of Data and Meta-data

Effective use of meta-data requires close integration with data not only in terms of representation, but also in computer processing. Such integration in turn requires software tools for co-management of data and meta-data. This section discusses several approaches to integration: self-describing files; software tools for self-describing files; data management, data dictionary, and statistical analysis systems; and a new generation of data analysis systems.

6.1. Self-describing Data Files

One approach to integration of data and meta-data is to combine the two in a single "self-describing" physical file on some type of computer storage media (tape, cards, disk, etc.). Such files contain two logical components: a data definition section and data section. The data definition section may be broken down into file level description and descriptions of individual fields, as illustrated above in exhibit 1. The data section consists of one or more records made up of fields as described by the data definition section (hence the term "self-describing").

The codata (common data) self-describing file format, for example, was developed by Lawrence Berkeley Laboratory in the 1970s to permit different data management, retrieval, and analysis modules to share data and meta-data without being constrained to a single physical data file format. In codata files, both data definition and data sections reside in a single physical file, with information stored in character representation within fixed-length logical records as defined in the data definition section. The primary virtue of the codata format is its simplicity. Since both description and data are simple ASCII text, both can be read, written, and modified using any text editor, text-oriented tools, or formated read and write statements from a programming language. Codata files are easy to read and understand, to transport between dissimilar computers, and to convert to other file formats. Most simple formated data files can be converted to codata format simply by prepending a hand-edited data definition section. Conversely, codata files can be used to produce input for programs like SPSS, SAS, and MINITAB that require different syntax conventions for their own self-describing input files.

Since 1980 there has been increasing interest in meta-data and self-describing files from the standpoint of both formal and de facto data interchange standards [Roussopoulos 1982]. The International Standards Organization (ISO) and American National Standards Institute (ANSI) are close to adoption of an ISO/ANSI standard self-describing file format [ISO 1984]. In the past few years, the DIF format used by Visicalc and many other popular microcomputer spreadsheet programs has become a de facto standard for data and meta-data used in such programs [Kalish and Mayer 1981, Software Arts 1981]. Organizations and individuals working in the area of computer aided design and computer aided manufacturing (CAD/CAM) are proposing to extend their Initial Graphics Exchange Specification (IGES) to make that data exchange format more self-descriptive [ANSI 1981, ANSI 1984b].

6.2. Self-Describing File Management Tools

Self-describing files provide a good starting point for integrating data and meta-data. Software tools for processing such files can use meta-data from the data definition section to locate, control and manipulate the data. In order to maintain the closest degree of integration between data, meta-data, and processing operations, such tools should be designed so that they operate on data and meta-data as a unit. That is, self-describing file management tools should use self-describing files as input and produce self-describing files as output.

The Codata Tools, developed at LBL for use with codata files, provide an operational example of software designed around this principle. They are a set of programs which read, write, and restructure the self-describing codata files discussed above [Merrill and McCarthy 1983]. These tools manipulate both data and meta-data, so that the output of any operation is itself a codata file. Individual codata tool programs perform different specific tasks, including extraction of specified rows and/or columns from a file, sorting a file, relational joins, tabulations aggregating on specified key values, and other operations. For example, exhibit 3 shows a schematic diagram of column (field) extraction using the codata tool called "cocol". Cocol extracts a specified subset of columns (fields) from the input codata file and puts them in a new codata file, complete with all file and field-level meta-data.

Exhibit 3: Column (Data Element) Selection Tool

Input: **file1**

nde = 5 areas = 3 missing = -999 etc....				
state	stub.geo	pop80	pop70	pop60
19	Iowa	2913808	2825368	2757537
48	Texas	14229191	11198655	9579677
49	Utah	1461037	1059273	-999

Operation: **cocol stub.geo pop60 pop80 <file1 >file2**

Output: **file2**

nde = 3 areas = 3 etc....		
stub.geo	pop60	pop80
Iowa	2757537	2913808
Texas	9579677	14229191
Utah	-999	1461037

The Codata Tools were designed and written in accordance with principles outlined by Kernighan and Plauger [Kernighan and Plauger 1976]. They are modular; each tool performs a specific limited task. They follow the UNIX and Software Tools conventions of standard input and output ("<" means read from the file name that follows and ">" means write onto the file name that follows). The output of one module can automatically serve as the input of another. As in UNIX, codata tool operations can be chained together sequentially using "pipes" ("|"). For example, in exhibit 3, instead of simply directing the standard output to file2 (">file2"), we could have specified "| cosort pop80 > file2" in order to have the records in file2 sorted in order of their 1980 population values.

Like codata files, the primary virtue of the codata tools is their simplicity. They provide basic relational operations such as select, project, and join, without the overhead of a full database management system. They use and produce standard human-readable ASCII files.

6.3. Data Management Systems

At the other end of the spectrum from the simplicity of the codata tools, database management systems provide a rich variety of capabilities for integration of data and meta-data. At present, however, none of the many different data management systems provide capabilities for user-defined extensions to their system-level meta-data. That meta-data still is typically limited to physical information necessary for data processing (e.g., names, data types, field sizes). Users can define and manage meta-data as standard database files. They may even be able to build in capabilities for linking data and meta-data. But it is more difficult, if not impossible, to extend the DBMS manipulation functions to operate concurrently on data and meta-data.

For example, relational data management systems, which inspired part of the functionality of the Codata Tools, provide set operations in which both inputs and outputs are relations. In current relational systems, however, it is not possible for users to add to the system-level meta-data that automatically is generated for new relations that result from standard select, project and join operations.

Both relational and non-relational data management systems provide other useful features for scientific data and meta-data. Most include concurrency control, backup, crash recovery, security and integrity mechanisms, indexing and query facilities. SIR can create and use system files including both data and meta-data for SAS, SPSS, and other statistical packages; it includes some types of meta-data, such as value label sets and multiple missing data codes, not usually found in data management systems [Fox 1984]. RIM supports two dimensional arrays and a variety of scientific operators [Boeing 1983]. On the other hand, most data management systems do not provide capabilities necessary for certain kinds of scientific data [McCarthy 1979, Shoshani 1982, Shoshani et al. 1984], and meta-data [McCarthy 1982].

In general, current data management systems have two important deficiencies from the standpoint of integrated management of data and meta-data. First, as discussed above, it is not possible to extend system-level meta-data. Second, few existing data management systems support the range of data types and structures (e.g., variable length text, multiply occurring objects, nested structures) that are most appropriate for meta-data representation, as outlined above in sections 3 and 5.

6.4. Data Dictionary Systems

Data dictionary systems, currently used primarily for administrative applications, offer another approach to integrated management of data and meta-data. Early data dictionary systems simply provided passive documentation of data elements, programs, procedures, etc., without active links to the data they described. Now, however, system developers are expanding dictionary contents and capabilities [Curtice 1981]. In more and more database systems, the data dictionary has become the central source for active definition, updates, and control of all database activities. In most such systems, meta-data is stored and maintained physically separate from the data it describes. Logical relationships between data and meta-data are defined in the data dictionary, however, so users can view them logically (if not operate on them) as self-describing files. The VAX/VMS System's Common Data Dictionary has extended the concept to provide a central source of meta-data for stand-alone programs as well as different database systems [Digital 1983].

From the standpoint of integrated management of data and meta-data, current data dictionary systems suffer from the same basic limitations as database management systems. Their data types and structures are not adequate for representing the full range of meta-data, and it is not easy to add new types of meta-data as required. In the future, however, new approaches may overcome at least some of these deficiencies [Sibley 1984, ANSI 1984a].

6.5. Statistical Analysis Systems

Many scientists use software packages such as SPSS [SPSS 1983], SAS [SAS 1982], P-STAT [Buhler 1979], and RS/1 [BBN 1983] to manage and analyze their data and such meta-data as those systems support. Most such systems provide some additional meta-data capabilities (e.g., value label sets, multiple missing data codes), as noted above in section 3. While adequate for many applications, statistical systems have their limitations for integrated management of data and meta-data. Most have evolved from single-user batch processing systems. Although they were among the first systems to support a non-procedural data definition for their input and <u>internal</u> self-describing files, they do not as a rule <u>produce</u> such files as a routine part of any given data manipulation or analysis procedure. Hence it is generally difficult, if not impossible, to make the output of one routine easily available as input for another. As with current data management systems, it is not possible to extend the types of meta-data supported by statistical systems. Although they provide a wide range of checking, manipulation, and analysis capabilities for numeric data, their facilities for management of complex textual meta-data structures are modest at best. Data definition syntax is usually idiosyncratic — part procedural and part nonprocedural. They do not handle concurrency control or security for multiple users.

6.6. Data Analysis Environments

Three of the most interesting meta-data developments in the area of statistical computing -- System S, the Analysis of Large Datasets (ALDS) Project, and the Language of Data Project — all are based on self-describing data structures. "S", an interactive computer system, language, and software environment for data analysis was developed at Bell Laboratories and is now available under license from AT&T [Becker and Chambers 1984]. As in Unix, all S functions read and write standard input and output which can be chained together or redirected to files. All inputs and outputs in S are self-describing data structures made up of discrete data and meta-data components. These self-describing data structures, which resemble LISP in many respects, are quite flexible and extensible. New functions and data structures can be defined by users, and only those portions of a data structure that a function uses need be defined for it. The Analysis of Large Data Sets Project (ALDS) at Pacific Northwest Laboratory is developing a prototype data analysis environment, including extensions to S, to provide statistical analysts with data structures that can capture the entire data analysis process [Nicholson 1984]. For the Language of Data Project, the basic self-describing data structure is a table [Dolby 1984]. As noted above, this project has sought to define a minimally sufficient set of meta-data that such tables ought to include (e.g., row and column headers and sub-headers, source citations, footnotes, etc.). It is also developing software to perform basic types of data manipulation and analysis functions on such self-describing tables.

7. Summary Conclusions

This paper has discussed types, uses, and representation of meta-data, with special emphasis on scientific applications. It has identified a variety of specific types of meta-data that are, or could be, especially useful to define, locate, and control scientific data. Many types of meta-data, unlike most scientific data, contain multiple occurring textual data structures and information inherited from one hierarchical level to another. Nevertheless, successful use of meta-data, like data, depends upon standardized representation. General principles for representation of meta-data are important regardless of type of data or degree of computer automation. Those general principles were summarized under three sets of three headings: _clarity_ (differentiation, structure, simplicity); _maintainability_ (integration, authority, parsimony); and _flexibility_ (transformability, recursive reference, extensibility).

In order to integrate their data and meta-data, scientists need database software that facilitates representation and management of both. Data dictionaries, database management systems, and statistical packages each provide useful facilities and approaches in this respect. Unfortunately, none of them currently offer sufficiently flexible and extensible meta-data capabilities. Future software should extend rather than limit what scientists can do with meta-data. In the meantime, careful attention to meta-data representation principles can help make use of existing software more productive.

Richer meta-data, more explicit attention to its representation, and better tools for its use can all contribute to improvements in scientific data quality and productivity. The problems will not be solved quickly, but many of them can be tackled incrementally. All of these tasks call for ideas and cooperation from practicing scientists as well as computer professionals.

Acknowledgments

This work was supported in part by the Office of Scientific and Technical Information and the Office of Energy Research of the U.S. Department of Energy, under Contract No. DE-AC03-76SF00098. Pat and Jim Kremer, Arie Shoshani, Burkhard Kolb, Frank Olken, and Harvard Holmes provided encouragement and valuable suggestions.

References

ANSI 1981

American National Standards Committee Y14.26, "Digital Representation for Communication of Product Definition Data," American Society of Mechanical Engineers, 1981.

ANSI 1984a

American National Standards Committee X3H4, "X3H4 Base Document," Information Resource Dictionary Systems (IRDS) Technical Committee DRAFT Version, July, 1984.

ANSI 1984b

American National Standards Committee Y14.26, "IGES Change Order Serial No. 232" (Add Tabular Data Property Form 11), draft copy, 1984.

BBN 1983

RS/1 User's Guide, BBN Research Systems, 1983

Becker and Chambers 1984

Becker, R.A. and J.M. Chambers, S: An Interactive Environment for Data Analysis and Graphics, Wadsworth, 1984.

Boeing 1983

Boeing Computer Services Company, BCS RIM — Relational Information Management System, Version 6.0 User Guide, Boeing Computer Services Company, 1983.

Boral 1982

Boral, H., D.J. DeWitt, and D. Bates. "A Framework for Research in Database Management for Statistical Analysis," Computer Sciences Technical Report #465. Madison: University of Wisconsin, Computer Sciences Department, February, 1982.

Buhler 1979

Buhler, S. and R. Buhler. P-STAT 78 User's Manual. Princeton, NJ: P-STAT, Inc., P.O. Box 285, 1979.

Curtice 1981

Curtice, R.M. "DATA DICTIONARIES: An assessment of Current Practice and Problems," Proceedings of the Seventh International Conference on Very Large Databases, (Cannes, France, September, 1982) (AC Order No. 471810).

Date 1984

Date, C.J., "Some Principles of Good Language Design, with Especial Reference to the Design of Database Languages," 14 SIGMOD Record 3 (November, 1984), pp. 1-7.

Digital 1983

Digital Equipment Corporation, VAX-11 Common Data Dictionary Language Reference Manual, Digital Equipment Corporation, Maynard MA 1983.

Dolby 1984

Dolby, J., "Put the Information in the Database Not the Program," this Proceedings, pp. 125-140.

Fox 1984

Fox, J.S., SIR/DBMS PRIMER, University of Wisconsin System, Madison Academic Computing Center, 1984.

Institute for Social Research 1973

Institute for Social Research, OSIRIS III, 6 vols, Ann Arbor: Institute for Social Research, University of Michigan, 1973.

ISO 1984

International Organization for Standardization, "Information Processing -- Specification for a data descriptive file for information exchange," draft international standard ISO/DIS 8211, International Organization for Standardization, 1984.

Kalish and Mayer, 1981

C.E. Kalish and M.F. Meyer, "DIF: A Format for Data Exchange between Applications Programs," BYTE Magazine (November, 1981), 174-206.

Kernighan and Plauger 1976

Kernighan, B.W., and P.J. Plauger, Software Tools, Reading, Mass: Addison-Wesley, 1976.

Klensin 1981

Klensin, J.C. and D.B. Yntema, "Beyond the package: A new approach to behavioral science computing,"20 Social Science Information 4/5, (1981), p. 787-815.

McCarthy 1979

McCarthy, J.L., "Data Characteristics, Application Requirements, and Database Management Tools: Matching Statistical User's Needs and Systems Capabilities," Proceedings of Computer Science and Statistics: 12th Annual Symposium on the Interface, University of Waterloo, Ontario, Canada, (May, 1979), p. 37-44.

McCarthy 1982a

McCarthy, J., "Enhancements to the CODATA Data Definition Language," Lawrence Berkeley Laboratory Technical Report, LBL-14083 (February 1982).

McCarthy 1982b

McCarthy, J., "Metadata Management for Large Statistical Databases," in Proceedings of the Eighth International Conference On Very Large Databases (Mexico City, September 1982), 234-243.

Merrill and McCarthy 1983

Merrill, D. and J. McCarthy, "Codata Tools: Portable Software for Self-Describing Data Files," in *Computer Science and Statistics: Fifteenth Annual Symposium on the Interface* (Houston, Texas, March 1983), 245-251.

Miller 1956

Miller, G.A., "The Magical Number Seven, Plus or Minus Two: Some Limits on our Capacity for Processing Information," *63 Psychological Review* (1956), p. 81-97.

Nicholson 1984

Nicholson, W., "Managing the Data Analysis Process," this *Proceedings*, pp. 141-166

Relational Technology 1984

Relational Technology, Inc., *INGRES Reference Manual*, Relational Technology, Inc. 1984.

Roussopoulos 1982

Roussopoulos, N., ed., *Proceedings of the Workshop on Self-Describing Data Structures*, University of Maryland, Oct. 27-28, 1982.

SAS 1982

SAS User's Guide: Statistics, SAS Institute Inc., 1982.

Schroeder et al. 1975

Schroeder, J.R., W.C. Kiefer, R.L. Guertin, W.J. Berman, "Stanford's Generalized Database System" in D.S. Kerr, ed., *Proceedings of the International Conference on Very Large Data Bases*, (Framingham, MA), September, 1975.

Shanks 1980

Shanks, M., "UNIX Software for Survey Analysis," University of California, Berkeley, Survey Research Center, 1980.

Shoshani 1982

Shoshani, A., "Statistical Databases: Characteristics, Problems, and Some Solutions," in Proceedings of the Tenth International Conference on Very Large Databases (VLDB), Mexico City, 1982, 208-222.

Shoshani et al. 1984

Shoshani, A., F. Olken, and H.K.T. Wong, "Characteristics of Scientific Databases," in Proceedings of the Tenth International Conference on Very Large Databases, (Singapore, September 1984).

Sibley 1984

Sibley, E.H., "An Expert Database System Architecture Based on an Active and Extensible Dictionary System," Proceedings of the First International Workshop on Expert Database Systems, (Kiawah Island, SC, October 1984), p. 566-585.

Smith 1978

Smith, J.M. and D.C.P. Smith, "Principles of Database Conceptual Design," Proceedings of the NYU Symposium of Database Design, (1978), p. 35-49.

Software Arts, 1981

Software Arts, Inc., "DIF Technical Specification," DIF Clearinghouse, P.O. Box 527, Cambridge, MA, 1981.

Sparr 1982

Sparr, T.M., "Units and Accuracy in Statistical Databases," in H.T.K. Wong, ed., Proceedings of the First LBL Workshop on Statistical Database Management, LBL-13851, March, 1982.

SPIRES 1973

SPIRES Staff, Design of SPIRES II, Vol. 1, 2nd ed. revised, SCIP/ Campus Computing Facility, Stanford University, July, 1973.

SPIRES 1983

SPIRES Staff, A Guide to Database Development — A SPIRES Primer, Stanford University, 1983.

SPSS 1983

SPSS, Inc., SPSS-X User's Guide, McGraw-Hill, 1983.

Stewart 1978

Stewart, D. and M. Seda, Data Processing in the National Health Insurance Study, RAND Corporation, 1978.

PUT THE INFORMATION IN THE DATABASE, NOT THE PROGRAM

James L. Dolby

San Jose State University

San Jose, California

PUT THE INFORMATION IN THE DATABASE, NOT THE PROGRAM*

James L. Dolby

San Jose State University

Introduction

Data analysis is, and always has been, inherently interactive. The analyst reads the data, through whatever window available, to learn what information they contain. Modifying the way the data are arranged and displayed makes different relationships more evident. Indeed, the technology of data analysis revolves around the search for new and different ways to represent to the analyst the information contained in data sets.

The utility of this technology depends critically on the time lag between the shaping of the request for a particular arrangement in the analyst's mind and the response to that request. Rapid turnaround permits the analyst to think in real time. Slow turnaround causes lapses in concentration similar to those that would occur if we were forced to read running text with a visit to the library after each paragraph or two.

It is not surprising, then, to find that data analysts have always been deeply involved in the development of mechanical aids that would speed up data analytic procedures. Hollerith was, after all, working for Census when he developed the first punched-card equipment, and for 50 years the primary use of that equipment was in processing data to report form, for ongoing analysis of operations in government and business.

*This work is part of the Language of Data Project, supported by a grant from the System Development Foundation.

High-speed general purpose computers speeded up the process, but with certain notable exceptions, did not change it in kind. Today's model of a database management system tied to a report writer provides rapid turnaround for the analyst if, and only if, the analyst has anticipated all the possible report forms that might be useful. Statistical packages impose the same kind of constraint. A new thought of how the information might be rearranged is necessarily followed by the "trip to the library" approach of writing a new program.

In order to obtain greater generality of programs, and hence fewer trips to the library, it is necessary to put more information in the database. The question is which information, and in what form. This paper discusses a seemingly innocent example — a small data set and the command TOTAL — as an illustration of how this problem might be approached.

An Example

We do not need to delve into the more sophisticated operations of data analysis to see where the problem lies. Consider, for instance, the data set in Figure 1, shown as it was represented in a standard relational database system for a microcomputer.

The database system has a potentially useful command, TOTAL, which can be exercised interactively. The system can differentiate between character strings and numeric strings, so that it can avoid the problem of trying to add character strings. It also allows for 8-character names for variables, to remind us of what the "data" mean (although it does not allow display of those names with the values). Beyond that, the user is on his or her own.

THE DATA

500100	3995	4.91	9.11	14659	0	0	0	.	.	.
500200	3731	6.73	37.90	13815	0	2	0	.	.	.
500300	1965	0.97	64.27	15503	0	0	0	.	.	.
500400	2188	1.46	16.59	18438	0	1	0	.	.	.
500500	4412	0.25	14.48	20000	0	0	0	.	.	.
500600	3086	1.72	23.49	15066	0	0	0	.	.	.
500700	1278	2.19	49.37	14018	0	0	0	.	.	.
500800	2476	2.14	55.13	7059	0	0	0	.	.	.
500900	4905	9.66	17.61	8461	0	0	0	.	.	.
501000	3967	5.55	48.12	9119	0	0	0	.	.	.
501100	5855	3.81	53.29	13228	0	1	0	.	.	.
501200	3659	4.26	56.74	13569	0	0	0	.	.	.
501300	4094	4.64	20.91	16065	0	0	0	.	.	.
501400	4117	4.25	50.86	11709	0	1	0	.	.	.
501500	5122	4.24	63.51	12647	0	1	0	.	.	.
501600	5336	6.11	26.12	10844	2	0	0	.	.	.
501700	4087	1.27	72.62	12637	0	0	0	.	.	.
501800	4666	1.46	54.80	16270	0	0	0	.	.	.
501900	1665	4.92	28.53	12180	0	0	0	.	.	.
502000	7083	3.13	32.19	12534	0	0	0	.	.	.
. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .

Figure 1. The data (from the Santa Clara County, California, Office of Management and Budget)

The limitations of this primitive representation become all too apparent as soon as we examine the data more closely to find what TOTAL should mean for each column in this example:

> The first column lists the census tract numbers for Santa Clara County, California, as defined by the Bureau of the Census for the 1980 census of the population. A proper response to the command TOTAL is "Santa Clara County."
>
> The second column lists the number of people in each tract as determined by Census. The proper response here is the sum of the numbers. However, this propriety is established by the facts that (a) the census tracts constitute a partition of Santa Clara County and (b) "number of" is additive over a partition.
>
> The third column lists the percentage of black people in each tract. Percentages within the tract are not additive over the tracts. In isolation, we might be satisfied with an average, or median, of these numbers. However, since the second column gives the population figures, we can multiply the given percentage by the population to find the (approximate) number of blacks in each tract, add the populations over all tracts, and then divide by the county population to find percent black for the county.
>
> The fourth column lists the percentage of hispanic people in each tract and would be treated exactly like the third column to find percent hispanic for the county.

The fifth column lists median income for each tract. Here the proper response to TOTAL is either a median of the medians or a weighted median of the medians. A rational system should choose one of these computations as a default, apply it, append a footnote to that effect, and provide for a simple means of changing the default.

The next three columns are the first 3 of 15 which give the number of people excused from jury duty by reason for excuse. These columns represent the focus of the assembled table, and as such, should be distinguished internally for several reasons, some of which are discussed below. Like the population column, they represent counts over a partition, and the proper response to TOTAL for each one is the numeric sum.

Clearly, if we are to support a command as "simple" as TOTAL in a rational fashion, we must have a fairly sophisticated system.

A Detail of Computation

The problem of computing percent black for the county illustrates two points. First, we need the capability of stating and storing in the system, arithmetic relations between values in the data: e.g., "Black population = % black times population" (within any geographic area). The second point is of particular importance in analyzing data provided by others. If we work with exact arithmetic, the process of determining the number of black people in, say, tract 500100 would involve the following computation:

Black population in tract 500100 = 3995 x .0491 = 196.1545

As people do not come in fractional units, we conclude that there were either 196 or 197 black people in the tract. Working backwards, we can calculate the percentages for both possibilities as

$$196/3995 = 0.049061326 = 4.91\%$$

$$197/3995 = 0.049311639 = 4.93\%$$

As the first result, 4.91%, agrees exactly with the starting figure, .0491 in the equation above, we can be reasonable sure that 196 is the correct population figure to use in the sum. This procedure can, of course, be repeated for every row to determine whether the computed sum for the column can be considered exact or in possible error due to rounding of the given percentages.

Of greater potential significance is that fact that, given the computations above, we clearly cannot get a result of 4.92%. Therefore, had such a figure appeared in the table, we would be forced to conclude that either the population figure or the percentage was in error. Were several such errors to appear, this might alert us to some systematic error in the data. For example, Census revises its estimates of population over time as more information becomes available. Hence it would not be unusual to find that the two columns of data had come from different Census reports, reflecting different stages of the long-term revision process.

Good auditors know that very small errors, which may not materially effect today's decisions, are sometimes the essential keys to deeper problems in the data. It is possible to program computers to check for such "details" routinely — if the computer is given the necessary information to do so.

Totals Across

Although the database system used supports totals down, even if in a primitive way, it does not support totals across — perhaps in simple recognition of the fact that adding things like "census tract number" to things like "population" would lead to silly answers. However, totals across are frequently useful. In this case there are two potential totals of interest:

> Percent black + percent hispanic. Percentage within the tract is additive over disjoint sets — but are the sets disjoint? In the Census publication General Population Characteristics — California, the counts for "persons of Spanish origin" are footnoted: "Persons of Spanish origin may be of any race". Hence summing these two columns across is inappropriate

> Total over the 15 reasons for excuse from jury service. In this case the reasons are disjoint, and counts are additive: hence totals are appropriate.

Indeed, totals over "reasons" within the tract, and totals over tract for each reason are probably the first things the analyst would want to look at. The role of the first five columns is to provide identification and permit rearrangement and regrouping of the "excuse counts," which are the focus of the data set. That is, these columns serve as "auxiliary variables", included in the collection to enhance the analysis of the "main variable", the excuse counts.

In fact, one way to look at this data collection is as a join of two tables, the first with the column heads

Tract	Population	% Black	% Hispanic	Median income

and the second with the column heads

	Reason for Excuse		
Tract	Noncitizen	Moved	- - -

with the join being made on the census tracts.

The first table can, in turn, be thought of as the join of four one-way tables, each with census tract in the stub. Structures of this sort are well anticipated in database systems and statistical packages. The second table is an ordinary two-way table of "number of people excused from jury service, by census tract and reason for excuse" (for selected time periods in Santa Clara County). Such structures are not naturally supported in database management systems, or for that matter in many statistical packages.

In addition to the ability to TOTAL ACROSS in a two-way table, we might also want to SORT ROWS BY TOTALS ACROSS or SORT COLUMNS BY TOTALS DOWN. We might want to TRANSPOSE the table — that is, interchange the stub and column heads. ALL these operations are naturally defined in terms of the two-way table, and hence are "unnatural" to systems which know not of this elementary structure.

Obviously, there is no need to suppress the four columns of auxiliary information in order to support the TOTAL and SORT commands — as long as the role of these columns is properly identified to the system. (It might be wise to suppress these variables for the TRANSPOSE operation in the interest of keeping the column-head structure as simple as possible.)

Displaying the Results of "TOTAL"

If the system is to be truly interactive, it must display the new table generated by the command — immediately. This requirement forces us to create a "table display algorithm" that will display any table. Such an algorithm would have to be based on a detailed description of the syntax of tables. However, since this syntax is inherent in the tabular form (see Dolby and Clark, 1982), it can be defined in terms of the structural elements of the table itself. Figure 2 identifies the basic structural pieces in a typical table.

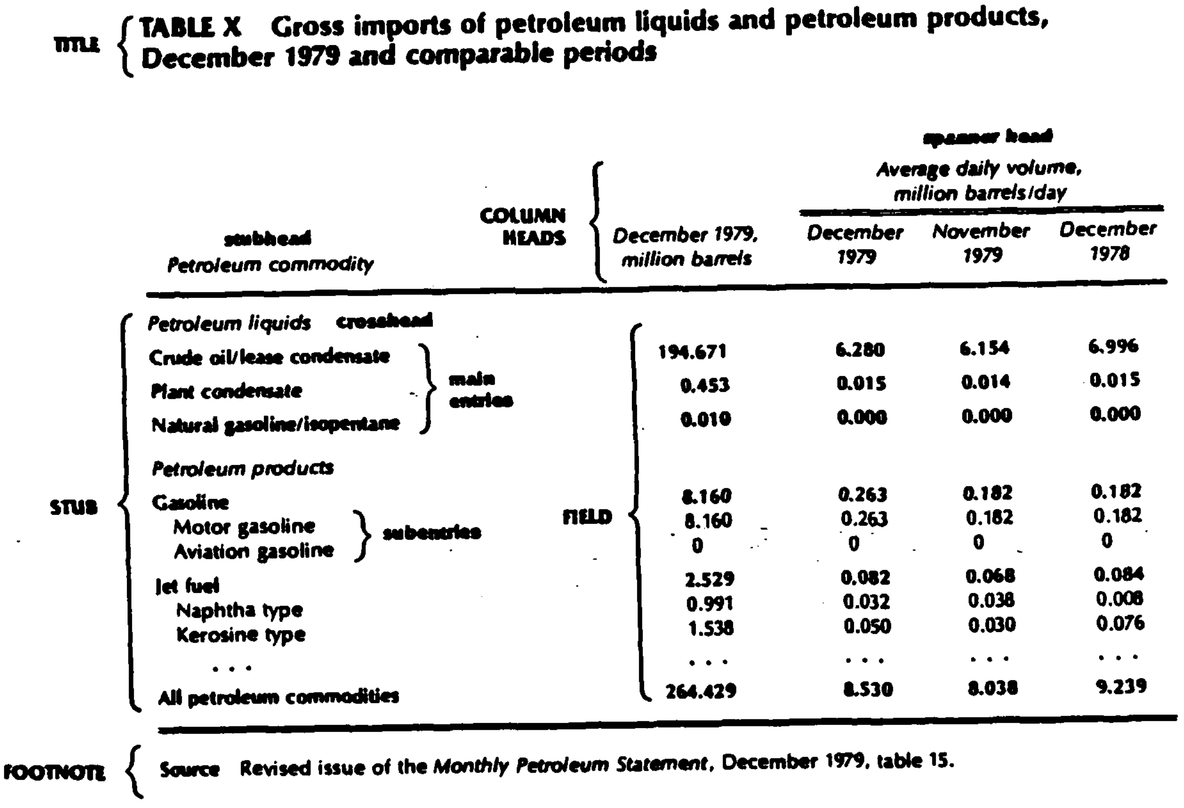


TABLE X Gross imports of petroleum liquids and petroleum products, December 1979 and comparable periods

Petroleum commodity	*December 1979, million barrels*	*Average daily volume, million barrels/day* *December 1979*	 *November 1979*	 *December 1978*
Petroleum liquids				
Crude oil/lease condensate	194.671	6.280	6.154	6.996
Plant condensate	0.453	0.015	0.014	0.015
Natural gasoline/isopentane	0.010	0.000	0.000	0.000
Petroleum products				
Gasoline	8.160	0.263	0.182	0.182
Motor gasoline	8.160	0.263	0.182	0.182
Aviation gasoline	0	0	0	0
Jet fuel	2.529	0.082	0.068	0.084
Naphtha type	0.991	0.032	0.038	0.008
Kerosine type	1.538	0.050	0.030	0.076
. . .	. . .	. . .	. . .	. . .
All petroleum commodities	264.429	8.530	8.038	9.239

Source Revised issue of the *Monthly Petroleum Statement*, December 1979, table 15.

Figure 2. The structural elements of a table (from Clark, 1984)

If the table is described to the computer system in terms of these elements, then every operation on the data must be viewed as an operation on the table which produces a new table. That is, the definition of the operation must include a definition of the effects of that operation on every element of the table.

At the display level the new row of values created by the command TOTAL would ordinarily appear below the original rows. Some government agencies prefer to display totals at the top of the table instead. In this context, however, the function of any visual display is to make the structure of the information visible (see Clark, 1983). Hence, in either position, the total row must be differentiated in some way from the other rows so that the totals can be distinguished from their components.

In a table with only a few columns this distinction is often carried by the wording of the stub entry, but in large data sets the stub is, in effect, out of visual range. The display algorithm for TOTAL therefore has to include some kind of typographic distinction, even if only a separation by space. This distinction is even more critical for subtotals within the table. In the census data in Figure 1, one of the purposes of including the four auxiliary variables is to support the computation of subtotals over different subsets, for a comparison of the excuse counts by population, percent black, percent hispanic, or median income in the various census tracts.

For example, the tracts might be grouped by median income according to whether the median for the tract is above or below the median income for the county. Subtotals for each of these groups could then be computed, provided the database included the statement of how the census tracts were partitioned into these two groups — that is, the "partition rule." Display of the new table, showing the subtotals, would then simply be a matter of reordering the rows by median income and including the subtotals immediately above each subset. Other "order rules" might also be used to rearrange the tracts within the two subsets.

Timing and History of Interactive Data Analysis

There are at least two more general considerations that should not be lost sight of. First, if a system is to be really interactive, it should let the user know when the requested computation is going to be more than a few seconds:

```
THIS COMPUTATION WILL TAKE 37 SECONDS
```

It should also give the user an opportunity to avoid longer computations:

```
THIS COMPUTATION WILL TAKE APPROXIMATELY 37 MINUTES:
DO YOU WISH TO ABORT, BATCH FOR LATER, OR CONTINUE?
```

Enough is known about the timing of algorithms to made such a capability easily available.

Second, it we have a computer system that responds quickly enough to support interactive analysis, then we must expect that it will be used to generate a large number of displays — some of which may be quite useful, others less so. The system should therefore be able to store the information needed to recreate any previous display on demand, along with some indication of how useful the display was thought to be at the time. (In a private communication John Tukey has suggested a 3-level utility code for this purpose.)

Fortunately the information needed to display a data set is much smaller than the data set itself. Moreover, successive displays often differ in rather simple ways, so that the differential display information is even smaller. Hence the problem of saving all display information should not be too difficult.

Types of Information Necessary to Support Interactive Display

The command TOTAL is not usually thought of as one of the topics in a course on advanced data analysis. On the contrary, it is so fundamental an operation that we have tended to take it for granted. However, the example above, though it is not exhaustive, is enough to show that we have to work a little harder if we are to support TOTAL as an interactive command. In particular, the following facts would have had to be included in the database to support totals down and totals across in this example:

1. "Census tracts" is a partition of the geographic entity "Santa Clara County, California".
2. "Reasons for excuse" is a partition of the set of "all reasons for excuse" from jury service.
3. "Black" and "hispanic" are overlapping subsets of people.

These statements refer to classificatory information. The first two statements could be stored as classification trees and referred to by tree name. The third fact could be deduced from the failure to find "black" and "hispanic" in the same tree. These trees would have to be documented by full bibliographic citations, giving the author, publication title, date, and so on. This information is necessary, for instance, to enable the user to identify different definitions of the census tracts for different censuses (see Dolby, 1983).

4. "Counts" are additive over partitions.

5. "Percentages" and "proportions" are additive over partitions of the whole on which they are defined.

6. "Percentages" and proportions" can be converted to counts if the counts for the whole are available.

7. TOTAL of medians can be defined in more than one way, and the preference must be stated.

The first two statements refer to applicability conditions for the operator TOTAL. The third statement is more an information requirement for the computation of counts. The fourth shows the need to state more than one definition of an operator, with the further requirement for a default choice.

8. The structural elements of the table must be defined to the system.

9. The variable displayed in the main field of the table, "number of people excused from jury service", must be identified by category name.

10. The variable listed in the table stub, "census tracts," must be identified by category name.

11. The variable represented by the main column heads, "reasons for excuse", must be identified by category name.

12. Auxiliary variables, such as "population", must be identified as such.

These statements all pertain to the syntactic structure of the table.

13. The results of a computation, such as the response to TOTAL, must be identified as such.

14. The partition rules used to form subtotals must be stated to the system.

15. The order rules used to establish row and column order must be stated to the system.

These statements are necessary to provide for rearrangement of the contents of the table at display time.

16. Timing algorithms must be provided for all nontrivial interactive computations.

17. Facilities for storing all display information must be supplied to allow re-creation of previous displays.

Such an enumeration raises an immediate question: can the costs of inputting this type of information be justified? The answer is fairly simple. If the machine is to respond properly, the information must be supplied. The question then reduces to one of how the information should be supplied -- as labels in a special program for a particular exercise, or in a more general way that allows both interactive response and the possibility of multiple use of related data sets. To this data analyst at least, the question begs the issue. To quote a recent television commercial, "You can pay me now, or you can pay me later".

The Beginnings of a System Design

What emerges from these considerations is the overall shape of a truly interactive data analysis system. Such a system must be able to store and retrieve not only tables of data, but chunks of information about those data in the form of trees, partition rules, order rules, applicability conditions for operators, and definitions of functions. It must also operate at a speed that allows real human interaction with the displays, warns the user about time-consuming computations, and provides a means of storing all displays — which are, after all, the analyst's working materials.

On the basis of preliminary experiments, it appears entirely feasible to construct such systems which will operate on rather modest computers.

References

Dolby, James L., and Nancy Clark (1982): The Language of Data, Los Altos, Calif.

Dolby, James L. (1983): Meaning from Data: Implications for Data Analysis and Database Management Systems, presented at the 1983 meeting of the American Association for the Advancement of Science, Detroit, May 28.

Clark, Nancy (1983): Statistical Presentation — of What, to Whom, and for Which Purpose? presented at the Joint Statistical Meetings, Toronto, August 18.

Clark, Nancy (1984): Tables as a Medium of Communication, presented at Online '84, San Francisco, October 29.

MANAGING THE DATA ANALYSIS PROCESS

W.L. Nicholson, P.J. Cowley, D.B. Carr, and M.A. Whiting

Pacific Northwest Laboratory

Richland, Washington

MANAGING THE DATA ANALYSIS PROCESS

W.L. Nicholson, P.J. Cowley, D.B. Carr, and M.A. Whiting

Pacific Northwest Laboratory

1. Introduction

The title "Managing the Data Analysis Process" probably has little meaning except for practicing data analysts. Such analysts understand that data analysis includes classes of activities such as data organization, data cleanup, data exploration, mathematical derivation, formal inference, and report writing. These activities generate information in forms that are difficult to keep organized. In fact a major portion of analysts' time is spent in organizing and keeping track of what has been done, what has been learned, and what should be done next to answer questions of interest. Even so, little has been done concerning the capture, management and restoration of information generated during a computerized data analysis. So, analysts have been forced to develop informal organizational devices. The few bright spots such as statistical packages that allow output from algorithms as input to other algorithms and have a diary capability are only small steps, in terms of the analysts' needs. Unfortunately with exceptions such as Tukey (1982) little has been written about managing the data analysis process.

This paper discusses the progress of an interdisciplinary team of data analysts and computer scientists working together on the Analysis of Large Data Sets Project (ALDS) (Nicholson et al. 1980) toward formulating the data analysis process and toward using a computer to manage that process. While the paper is basically philosophical we discuss implementation of some of the concepts in a prototype data analysis management software system. Data analysis management is a recent focus of ALDS software development; however, for several years we have been thinking about the topic and how it should be implemented. Carr (1979) and Nicholson (1983) have discussed data analysis management in the context of analyzing very large data collections and suggest that the frequency and breadth of such analyses have been severely limited because of the lack of organizational software.

Aspects of data storage and manipulation are discussed by Thomas (1983), Littlefield and Cowley (1983) and Littlefield, Cowley and Whiting (1984). While the terminology differs many of the concepts discussed here appear in crystalized form in Carr et al. (1984) and Nicholson et al. (1984).

Data analysis, and in particular, data analysis involving large complex data bases, is a difficult, cumbersome and time consuming task. The initial stage of such analysis often involves data organization and data cleanup. In concert with organization and cleanup, the insightful portion of the analysis continues as an iterative, clue-driven process. Current results suggest new analyses which, when performed, generate more data structures and more results. As the process evolves, a network of analysis paths and intermediate result nodes is generated. Some paths lead to dead ends, and some to further branching. The result can be chaos without substantial analyst-generated housekeeping. Specifically, the analyst must

- Deduce mnemonic names for data structures which both identify their location in the network and suggest their content.
- Clean up, organize and store command streams which define analysis paths.

- Organize hard copy, which may be graphical and tabular summaries of analysis results, or ideas concerning the meaning of results and/or directions for further analysis, and associate relevant hard copy with data structures.
- Maintain a data analysis overview which describes the analysis evolution and saved data structures.

This organizational and housekeeping task occupies a major portion of the total time spent by the analyst. Hence the time required for data analysis management limits the quality of the final product. Efficient use of analyst time necessitates computerizing this data analysis management process. Software tasks are needed that automatically

- Maintain descriptions and locations of data structures.
- Preserve all needed information to define the state of the analysis at particular times.
- Manage command/log files.
- Provide for viewing the overall data analysis process.
- Restore an analysis to any specified state.
- Utilize system resources effectively.

In concept, this type of data analysis management is well within capabilities of current computer systems. In fact, it just has not been addressed by software developers.

A viable data analysis management system provides important benefits in addition to allowing the analyst to make better use of time. These benefits result from providing a complete and accurate record of the data analysis. A first benefit is quality assurance. In data analysis many things may go wrong: erroneous conclusions may be drawn, inappropriate procedures may be used, algorithms may suddenly fail for particular data sets, and occasionally a wrong version of data may be used. This makes quality assurance exceedingly difficult; however, maintenance of organized documentation is one of the first steps toward quality assurance.

A second benefit is quality assessment. Tukey (1983) discussed the significance of capturing the data analysis process and proclaimed a need for "automated cartography of exploration" in order to assess the quality of data analysis. Specifically, he stated:

> We can review our own work looking for missed opportunities — opportunities perhaps to try new branches, perhaps make new syntheses. We can understand much better what others have done in their analyses, judging the completeness of their attacks and the adequacy of their syntheses.

A third benefit is pedagogical. A detailed record of what skilled data analysts do, including their blind forays, is valuable material for those interested in studying the data analysis process. Understanding the process helps us define computer tools which will improve our ability to perform data analysis in the future. Being able to articulate the process is important for teaching others how to perform data analysis.

2. Data Analysis Management

In order to manage the data analysis process through computerized data analysis management, we think of data analysis as a process consisting of computation, storage, and direction with the latter controlled by an analyst whose understanding of the data increases as the analysis evolves. The whole process is dynamic with time-dependent interlinkage between these three components. Figure 1 is a schematic of the data analysis process which appears as a path in the interior of a triangular prism. The process is described in terms of a three-variable barycentric crossed with time coordinate system. For each temporal cross-section, the status of the process is a mixture of computation, storage and direction. The record of the data analysis process is kept on three planar appendages to the prism, each containing a time history of a specific component. The record of the computation component consists of software utilization by time. Software includes both statistical/database management packages and runstreams of commands. In the long term, software systems may not maintain temporal compatibility. Software may become obsolete and whole computing systems may change. Thus for long time frames it may not be practical to maintain usable software. Nonetheless for good understanding of why the computation was done the way it was, the record of the computational component is critical. The record of the storage component consists of a directory of data structures as a function of creation time. The record of the direction component consists of analyst perception of the analysis and insight into the data as a function of time. Recording analysts' ideas needs to be easy to do because we wish the recording process to have minimum impact on the analyst.

An adequate record not only contains the above components but also the time-dependent interlinkage between them. Thus, software must be associated with input and output data structures. Perception and insight must be associated with specific data structures and/or command runstreams which initiated the ideas. Milestones in the process must be associated with the computation record which generated one from the other.

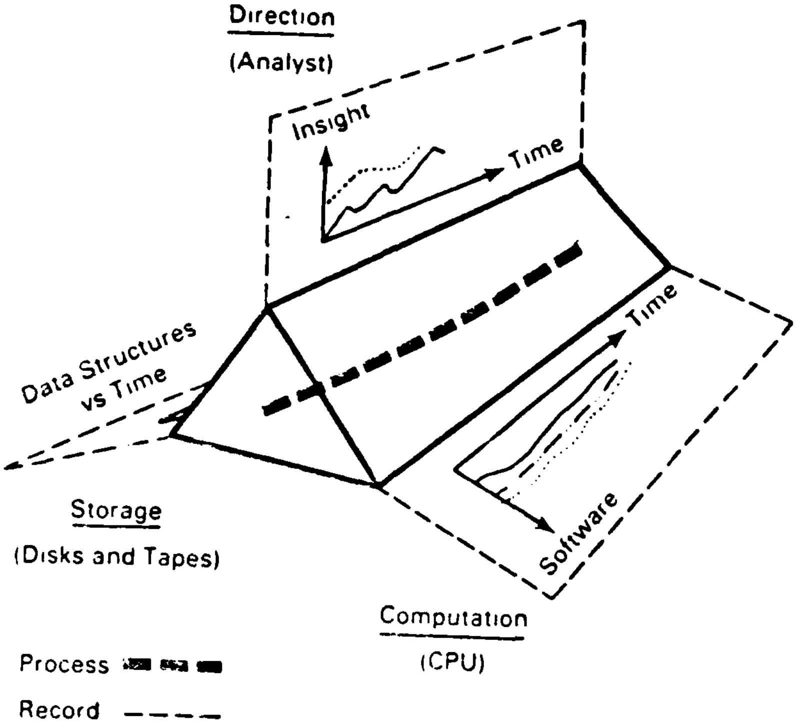


Figure 1. Schematic of data analysis process showing progress of analysis as dotted curve in interior of prism. Progress is archived as temporal records of three components — computation, storage and direction.

The purposes of data analysis management as we envision it are as follows:

- To maintain an overview (an abridged record) of the three components by time process.
- To manage temporal cross-section information.
- To preserve the details of specific temporal cross sections as milestones.
- To provide (within reason) the capability to return to any particular milestone in the process.

- To provide for editing and pruning so that the permanent computational record is an efficient statement of how successive milestones evolved from earlier ones.

In this section we describe our approach to handling such activities in a data analysis management system.

2.1 Data Analysis Save-States

Central to any data analysis management software must be the ability to capture, possible for permanent retention, the complete status of the data analysis at any time specified by the analyst. If the status is complicated, the capture process may, at least in part, be "virtual" in the sense that what is saved is the ability to recreate a particular status from an earlier status, as opposed to saving the particular status itself. The decision as to what portion of the statue is virtual should be determined by the software as a tradeoff between resources required to store the status and resources required to recreate the status. The status of a data analysis includes input data structures, output data structures, and software which produced the output from the input. The status of the analysis also includes characteristics of the process (analyst name, date, time of day, length of analysis session, etc.), and metadata describing the data per se (source, measurement units, how generated, missing value codes, etc.). Disk storage of this type of information is certainly not novel. However, there is more to the status of a data analysis; for example, part of the results may be graphics. There may be "doodles," mathematical observations and "aha's" that ordinarily would be jotted down by the analyst in the margin of printer output or on graphics hard copy. Thus, status includes a combination of CPU- and light-pen- (or similar interactive device) generated displays. Further, the status includes a list of special-purpose subroutines and/or macros invented by the analyst for the particular analysis. The analyst may record observations in written form as comments. Such comments are part of the status. Finally, the analyst may talk over, either with himself or with other analysts, specific aspects of the results and their implication. The discussion typically concerns answers to questions being sought from the data or directions for further analysis. Thus, audio record is also part of the status. Capturing the data analysis status means preserving all of these aspects of the status in a restorable form, which we call a "save-state."

2.2 Data Analysis Paths

As indicated above, the save-state as the status of the data analysis at any instant of time includes the runstream of commands which produced any desired data structures. Even though it is associated with the save-state, the runstream of commands needs to be considered independent of the rest of the save-state. Since much of data analysis is trial and error (with a large fraction being error), it is ridiculous in most situations to preserve the complete runstream between adjacent save-states. Exceptions to this policy include at least preserving how data analysis actually happened and cases where likely productive analyses turned out to be worthless. Usually, major editing of the runstream is necessary to reduce the set of commands to those necessary for the computation. Paths in the data an lysis process are these possibly edited runstreams of commands that connect adjacent save-states.

2.3 Data Analysis Networks

With the capability to store data analysis save-states and data analysis runstreams of commands, we have the units for describing the data analysis process as a network consisting of nodes (the save-states) with each node linked by paths (the runstreams of commands) to one or more other nodes. The network structure is the key for data analysis management and it serves several functions.

First, a display of the network structure is a visual summary of the entire data analysis process. Individual paths or sets of sequentially linked paths identify separate subprocesses evolving from particular data structures. Node identifiers allow anyone familiar with the analysis to locate that portion of the process of interest and to zoom in and view it in more detail.

Second, the network structure of paths and nodes organizes the runstream of all commands used in the process and all of the archived intermediate stages of the process in a cohesive manner so that commands are quickly associated with input and output data structures. This association provides an ordering so that the network can be simplified by removing intermediated nodes and joining runstreams at removal points.

Third, the network nodes provide entrance points so the analyst can look at archived results, pick up the analysis at a termination point or go back and restore an earlier status of the analysis either to check out details or to start a new analysis approach.

2.4 Interactive Functions

To utilize the data analysis save-state as a tool for data analysis management, the analyst must be able to create, restore and delete particular save-states. "Creation" links all components of the data analysis at a particular moment in time. "Restoration" recreates the status of the system at the time the specified save-state was created. "Deletion" erases all data structures and runstreams of commands which do not affect restoration of other existing save-states. For the last node in a tree structure this deletion is straight-forward; everything is deleted back to the previous save-state. If the save-state is an interior node in a tree structure deletion means deleting the data structures containing results and patching together pre- and post-node command runstreams so that nodes can be created from once-removed predecessor nodes. In a network the deletion process is more complicated.

2.5 Analyst Ideas

As indicated above, part of the status of the data analysis which should be preserved in the save-state are written and verbal comments by the analyst. These comments are documentation that describes the evolution of the analyst's perception of how the analysis should go and/or his insight into the process underlying the date. Oral comments can be stored on audio tape using a standard recording device. The key to usefulness of such audio information is automatic association with the save-state and quick recall when the save-state is restored. The audio record is particularly helpful to analysts who are verbally oriented. For them, more information can be dictated in a given amount of time than can be typed. With present audio equipment, the audio record portion of the save-state can be recalled in a semi-automatic fashion. The analyst must mount a tape selected by ID information stored in the save-state. The software then advances the tape to the beginning of the audio record for that save-state using an index stored in the save-state at the time the audio record was created.

"Doodles" and written comments on graphics displays may be written with a cursor and captured as an additional part of the graphics display. As part of the display, they appear automatically when the display is restored. If storage space is not a problem, software and/or cursor-generated displays may be archived directly in bitmap form. If otherwise, archival is virtual with storage of graphics-device-state parameters, plot commands and cursor-generated commands. Storage and recall of analyst annotation on printer displays is more complicated; at the present time, it is still an open question.

3. Prototype Implementation

A prototype system for data analysis management is being developed on the ALDS Project using a DEC VAX 11/780. The underlying software is EUNICE (which provides a UNIX-like environment while using the standard VAX operating system) and S [a statistical software package developed at AT&T Bell Laboratories (Becker and Chambers 1984)]. Changes and additions to S provide flexible interactive multi-dimensional graphics, capability for efficient handling of large data structures, and data analysis management tools. Graphics software is an improved version of that embedded in the MINITAB-based (Ryan et al. 1981) first-generation ALDS system. Previous ALDS graphics capability is discussed by

Littlefield (1982), Carr and Littlefield (1983), Nicholson and Littlefield (1983) and Carr and Nicholson (1984). The graphics output device is a RAMTEK 9400 with a tablet controlling cursor input. The data analysis management software is implemented as an S command that allows options to be specified graphically using a network display of the data analysis and cursor control of menu-type commands.

To facilitate description of our prototype system, we introduce a simple example of data analysis. An overview of that analysis, which involves five milestones preserved as save-states is illustrated in Figure 2. These save-states and the connecting analysis paths form the network representation of the data analysis process. In this case the network is a simple tree. The trunk of the tree is the data structure RAW.DAT which contains the original data as presented to the analyst. Icons are used in the display to denote the type of information in the save-state. (Because of limited resolution, in the prototype implementation icons are introduced only after the analyst has zoomed in on a particular save-state.) This raw data save-state consists only of a data structure which is identified by the file icon. The first step of the analysis was to generate a number of pair-wise plots for subsets of the data. The analyst studied these plots and found a number of data points that appeared discrepant. Some were found to be typos and were corrected. Others were unexplained anomalies and were set aside for further study after the analysis of the main body of data was completed. The analyst recorded his treatment of discrepant data as oral comments. The save-state designated RAW.DAT.PLT is the status of the analysis process at the completion of this data-screening phase.

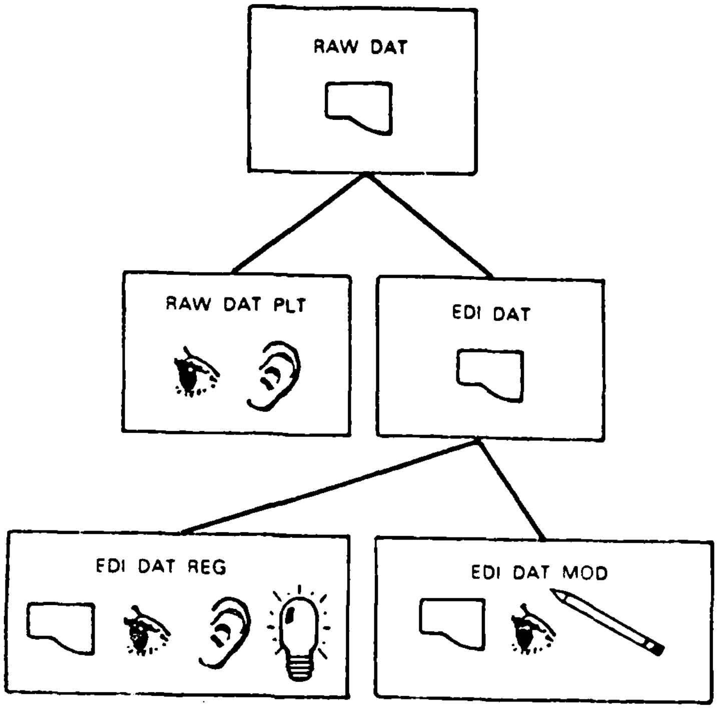


Figure 2. Overview display of data analysis preserved as a simple tree of five save-states and connecting analysis paths.

The eye and ear icons indicate that the derived results consist of plots and recorded oral comments. The second step of the analysis was to form an edited data structure based on the analyst's evaluation of the plots. The edited data constitute the save-state EDI.DAT which is identified by the file icon as containing only (derived) data structures. The third stage of the analysis was to fit several regressions to develop a preliminary understanding of the interrelationships in the data. The analyst studied the regression fits, residual plots and regression diagnostics. He formed conclusions and decided that a particular nonlinear model would account for the structure. He dictated these conclusions and a detailed statement defining the model and how the model should be fitted to the edited data. This regression and evaluation portion of the analysis was recorded as save-state EDI.DAT.REG. The icons indicate that the save-state includes data structures, plots, audio comments, and key insightful information. The latter is identified with a light bulb icon. The fourth stage of the analysis was to fit the nonlinear model to the edited data. Diagnostic displays and draft figures for a final report were generated. The analyst reviewed this information and added written comments in the margins of the nonlinear fit output and on the diagnostic plots. At this point the analysis was terminated. The final status constituted the save-state EDI.DAT.MOD. The pencil icon denotes that the save-state includes written comments.

The titles of save-states do not have to follow the hierarchical format in this example. In addition other information, such as author, and creation data, helps identify specific save-states.

In the prototype a variety of operations can be performed on save-states and command runstreams. These are presented in menu form and appear in windows overlaying the network diagram. Table 1 shows the nine menus of our prototype. A help feature provides online description of the menu items.

To create a save-state when a particularly interesting or important point has been reached, the analyst selects menu items using a data tablet. When the analyst chooses the state menu and then the CREATE command, many system activities are initiated. These activities include storing data structure names, macro names, plot names, printable file names, analysis time, author's name, and other bookkeeping information. The create menu provides for analyst-supplied annotation of the save-state and the inclusion of data from nonancestral nodes. Annotation includes the save-state title which appears in the node, icons that indicate what types of information the save-state contains, and comments entered by selecting the comments menu item. Comments may be typed or spoken. The system maintains typed comments in data files and spoken comments on computer-controlled cassette tape. Retrieval of comments for review and/or editing is performed by the system. Once finished with the save-state, the system queries the analyst about the command runstream which produced the just-stored derived results from data structures in the previous save-state. Default is to store the entire runstream of S commands; however, the analyst may edit the commands prior to storage.

Rather than explaining each of the menu options in this paper, a few options will be described in the context of the data analysis displayed in Figure 2. After the EDI.DAT save-state was created, after further analysis was done but prior to the creation of the EDI.DAT.REG save-state the network structure display is as shown in Figure 3. The highlighted path indicates the runstream of commands which produced the current analysis from the status of the system as of save-state EDI.DAT. (Highlighting is done as bold in figures and as a different color in prototype displays). The asterisk indicates that some derived data structures have resulted from that analysis; i.e., that new information exists which could be used to create a save-state distinct from EDI.DAT.

TABLE 1. Data Analysis Management Menus

Main Menu

- State (Node)
- Log (Path)
- Move Window
- Help
- S-Mode

State Menu

- Scan
- Restore
- Modify
- Show Network
- Erase Network
- Create
- Delete
- Move Windows
- Help
- S-Mode
- Return

Log Menu

- Scan Log
- Scan Plots
- Edit Log
- Create Macro
- Move Window
- Help
- S-Mode
- Return

Move Window Menu

- Menu Box
- Help Box
- State Box
- Comments Box
- Log Box
- Datasets Box
- Prompt Box
- Help on Move
- Return

Scan Menu

- View Comments
- Play Comments
- Scan Log
- Move Window
- Help
- Return

Modify Menu

- Modify Title
- Modify Author
- Turn Off/On Icons
- Datasets
- Comments
- Move Window
- Help
- S-Mode
- Return

Create Menu

- Modify Title
- Modify Author
- Turn Off/On Icons
- Datasets
- Comments
- Move Window
- Help
- Store/Return
- Quit/Return

Datasets Menu

- Add Datasets
- Delete Datasets
- Show Datasets
- Show Database
- Move Window
- Help
- Return

Comments Menu

- View Written
- Play Verbal
- Record Verbal
- Add Written
- Edit Written
- Move Window
- Help
- Return
- *(Advanced Audio control)

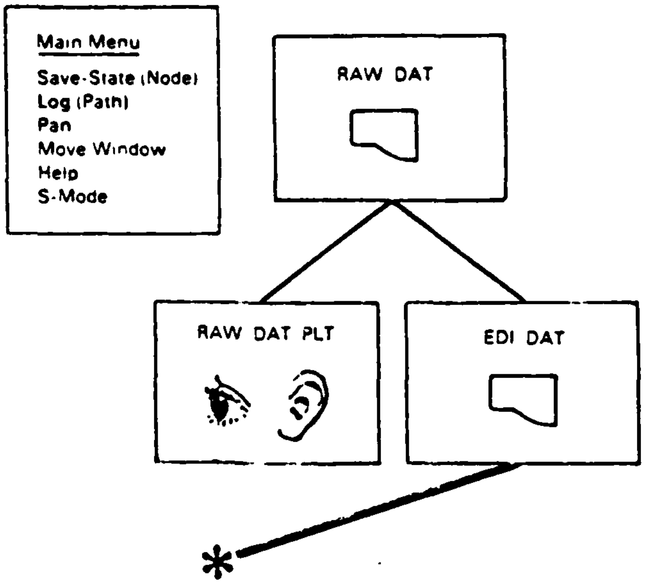


Figure 3. Overview display of data analysis after further analysis using EDI.DAT data structures but prior to creation of EDI.DAT.REG save-state. Highlighted (bold) analysis path and asterisk show that results are available to create a save-state distinct from EDI.DAT.

From the status of the analysis in Figure 3, the analyst created the save-state EDI.DAT.REG. Analysis results, files and commands defining plots were automatically included in the save-state. Using the SPEAK command the analyst also included oral comments in the save-state. Finally, he initiated the lightbulb icon option because the comments described a breakthrough in his understanding of the structure in the data. When the analyst finished annotating the new save-state and returned to the state menu, the new node was added to the network. The display of the data analysis process is shown in Figure 4. The new save-state as the current state of the data analysis is highlighted in this display.

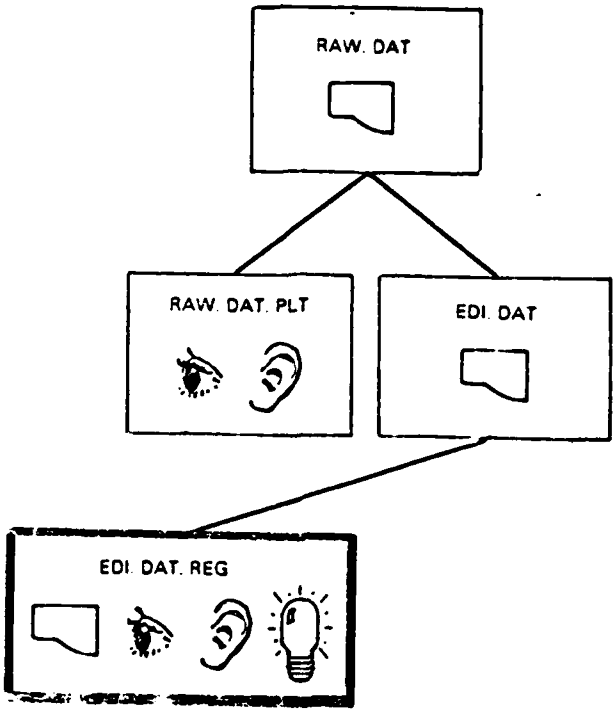
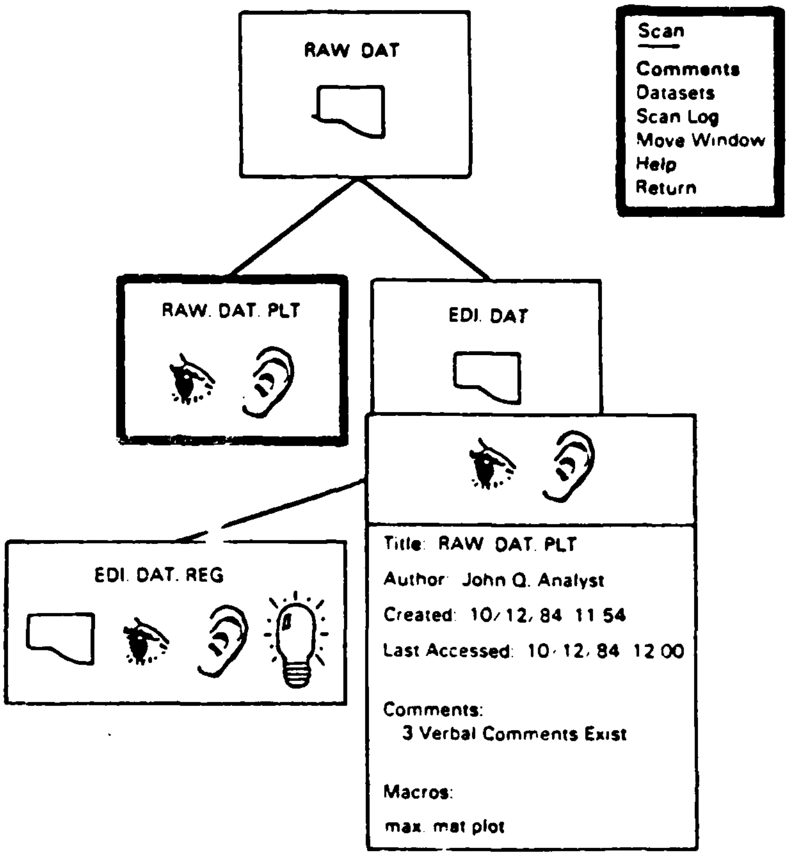


Figure 4. Overview display of data analysis immediately after creation of the highlighted (bold) save-state EDI.REG.DAT.

At this point after the archiving of the regression analysis, which is documented as part of save-state EDI.DAT.REG, the analyst wanted to investigate the anomalous data that were not used in the regression analysis. To do this the SCAN command on the state menu was selected, the save-state RAW.DAT.PLT was designated with a cursor and the COMMENTS command was selected from the scan menu. At that point, the status of the data analysis overview display appears in Figure 5. The scan mean appears in the window on the right side of the display. The highlighted save-state RAW.DAT.PLT and COMMENTS command in the scan menu show current status of analyst interaction with the display. Note the two icons indicating plots and oral comments are saved. The ID information includes title, author, creation date and time, and access date and



Title: RAW DAT. PLT

Author: John Q. Analyst

Created: 10/12/84 11 54

Last Accessed: 10/12/84 12 00

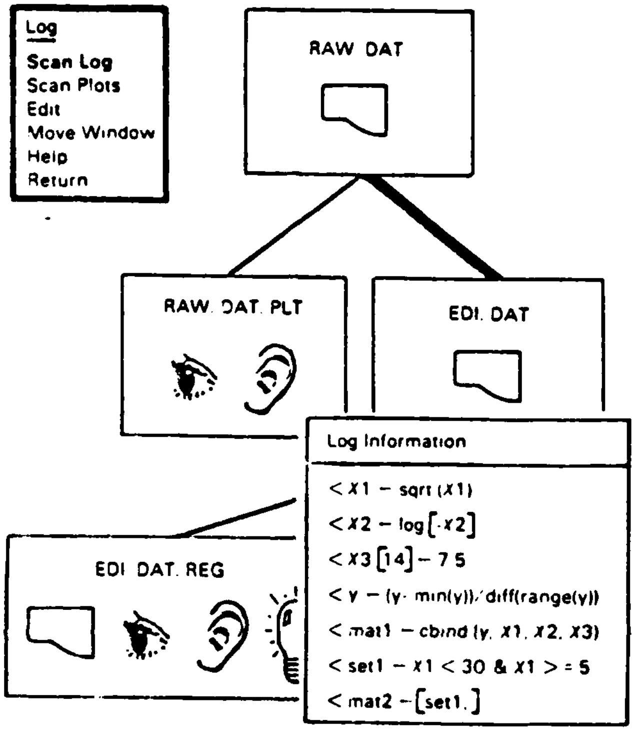
Comments:
3 Verbal Comments Exist

Macros:

max mat plot

Figure 5. Overview display of data analysis showing scan menu for operation on highlighted (bold) save-state RAW.DAT.PLT. Scanning view of save-state appears in window on lower-right. Selection of highlighted COMMENTS command will initiate replay of audio comments.

time. The analyst retrieved the oral comments which noted that the definition of the anomalous data was in the runstream of commands which created the edited data set from the raw data set. The analyst selected the LOG command in the scan menu and the SCAN LOG command from the log menu. He stopped the scan process at the point where specific data values were identified. The status of the display appears in Figure 6.



```
< X1 - sqrt (X1)
< X2 - log[·X2]
< X3[14] - 7 5
< y - (y- min(y))/diff(range(y))
< mat1 - cbind (y, X1, X2, X3)
< set1 - X1 < 30 & X1 > = 5
< mat2 -[set1,]
```

Figure 6. Overview display of data analysis showing log menu for operation on highlighted (bold) analysis path. A section of the command log appears in the window in the lower right. Scrolling allows detailed viewing of the entire log.

The analyst noted the definition of anomalous data, returned to the scan menu, then to the state menu, created the EDI.DAT.REG save-state and entered the S mode. He selected the anomalous data from the raw data file, compared the values with the fitted functional form and found that when viewed in a full structure context, the data were still anomalous.

5. Extensions

A variety of extensions to the prototype system are anticipated. Several extensions will provide more sophisticated methods for revising logs and the corresponding data structures. Interactive computing often yields errors and false starts that are reflected in the log and in data structures. In the current system analysts can correct both, but have little assistance from the system. The system should help in verifying the correctness of the revised log. If the computational effort is not too great, this could be done by executing the revised log and by comparing new data structures against old data structures that are known to be correct. To avoid accidental deletion of desired data, a "garbage collector" could be invoked to delete (or find) data not referenced in the revised log. If superfluous data structures were removed first, theorem provers could be used to remove unnecessary steps that lead from data structures to derived data structures. A tool to indicate where data were referenced would help in determining what needs to be done when alternative data are to be used. Thus, several tools exploiting the tight connection between log segments and data structures are under consideration.

Extensions to reduce storage requirements are planned. While the future is bright in terms of storage, in our finite storage computing environment, compact storage is advantageous. Frequently, only the log needs to be stored and data can be regenerated as necessary. Rather than storing data sets that are almost alike, differential file techniques (Aghili and Severance 1982) could be used to describe one data set in terms of another.

Most of the planned extensions involve shifting of tasks from the analyst to the computer. Some work will involve providing additional annotation capabilities. For example, plots and sketches drawn with a graphical input device will be stored, labeled, and retrieved. This will allow the hand annotation of any material to be stored and overlayed as desired. Also, work with different types of interaction devices is expected. For example, we are currently adding a track ball and toggle switches to the prototype system. Attention to interaction aspects between humans and computers is important in encouraging system use.

Our data analysis management system is an example of spatial data management (Friedell, Barnett and Kramlich 1982), the technique of accessing and organizing information via its graphical representation in an organized spatial framework. This technique is applicable in many settings. Providing visual organization to entities such as spoken comments and sketches will prove useful in a variety of contexts.

6. Summary

Advances in statistical computing have allowed data analysts to handle increasingly complex problems. The complex data analyses that are required with complex problems proliferate derived data structures and create evolving analysis states. Managing the analysis is typically a major task for data analysts. We have designed a system that helps the data analyst manage the analysis process through use of data analysis save-states. The system, which is interfaced to S, also makes use of graphical interaction, provides organization of command logs and encourages extensive annotation. While we are still in the process of evaluating our first prototype, it is clear that this collaborative effort of data analysts and computer scientists is making progress toward freeing the data analyst from much mundane but necessary organizational bookkeeping.

7. Acknowledgment

The ALDS program and, in particular, the research reported here is supported by the United States Department of Energy, Applied Mathematical Sciences Research Program, under contract DE-AC06-76RLO, 1830.

8. References

Aghili, H., and D.G. Severance. 1982. "A Practical Guide to the Design of Differential Files for Recovery of On-Line Data Basis." ACM Transactions on Data Base Systems 7(4):540-565.

Becker, R.A., and J.M. Chambers. 1984. "S"--An Interactive Environment for Data Analysis and Graphics. Wadsworth, Inc., Belmont, California.

Becker, R.A. and J.M. Chambers. 1984. "Design of the S System for Data Analysis." Communications of the ACM 27(5):486-495.

Carr, D.B. 1979. "The Many Facets of Large." In Proceedings of the 1979 DOE Statistical Symposium. Gattlingburg, Tennessee. CONF-791016, Oak Ridge National Laboratory, Oak Ridge, Tennessee. pp. 201-204.

Carr, D.B., and R.J. Littlefield. 1983. "Color Anaglyph Stereo Scatterplots -- Construction Details." In Computer Science and Statistics: Proceedings of the 15th Symposium on the Interface. North Holland Publishing Co., New York, New York, pp. 295-299.

Carr, D.B., and W.L. Nicholson. 1984. "Graphical Interaction Tools for Multiple 2- and 3-Dimensional Scatterplots." In Computer Graphics '84: Proceedings of the 5th Annual Conference and Exposition of the National Computer Graphics Association, Inc. Exposition of the National Computer Graphics Association, Inc. ISBN 0-941514-05-6, Vol. 2, Anaheim, California.

Carr, D.B., P.J. Cowley, M.A. Whiting, and W.L. Nicholson. 1984. "Organization Tools for Data Analysis Environments." In Proceedings of the Statistical Computing Section, American Statistical Association, Washington, D.C. pp. 214-218.

Friedell, M., J. Barnett, and D. Kramlich. 1982. "Context-Sensitive, Graphic Presentation of Information." Computer Graphics 16(4):181-188.

Littlefield, R.J. 1982. "Stereo and Motion in the Display of 3-D Scatterplots." In Proceedings of the 8th Annual Computer Graphics Conference, Engineering Society of Detroit, Detroit, Michigan, pp. 13-17.

Littlefield, R.J., and P.J. Cowley. 1983. "Some Statistical Data Base Requirements for the Analysis of Large Data Sets." In Computer Science and Statistics: Proceedings of the Fifteenth Symposium on the Interface. Houston, Texas. pp. 24-30.

Littlefield, R.J., P.J. Cowley and M.A. Whiting. 1984. "Some Data Manipulation Requirements for Exploratory Analysis of Large Data Sets." In Proceedings of the 9th International Codata Conference, Jerusalem, Israel. North Holland Publishing Company, New York, New York.

Nicholson, W.L. 1983. "Analyzing Large Data Sets: A Challenge for Statistical Computing." In *Proceedings of the Statistical Computing Section*. American Statistical Association, Washington, D.C., pp. 194-199.

Nicholson, W.L., D.B. Carr and D.L. Hall. 1980. "The Analysis of Large Data Sets." In *Proceedings of the Statistical Computing Section*. American Statistical Association, Washington, D.C. pp. 59-65.

Nicholson, W.L., and R.J. Littlefield. 1983. "Interactive Color Graphics for Multivariate Data." In *Computer Science and Statistics: Proceedings of the 14th Symposium on the Interface*. Troy, New York. Springer-Verlag, New York, New York, pp. 211-214.

Nicholson, W.L., D.B. Carr, P.J. Cowley, and M.A. Whiting. 1984. "The Role of Environments in Managing Data Analysis." In *Proceedings of the Statistical Computing Section*. American Statistical Association, Washington, D.C. pp. 80-84.

Ryan, T.A., B.L. Joiner and B.F. Ryan. 1981. *MINITAB Reference Manual*. Duxbury Press, Boston, Massachusetts.

Thomas, J.J. 1983. "A User Interaction Model for Manipulation of Large Data Sets." In *Computer Science and Statistics: Proceedings of the 14th Symposium on the Interface*. Troy, New York. Springer-Verlag, New York, New York, pp. 118-128.

Tukey, J.W. 1982. "Control and Stash Philosophy for Two-Handed, Flexible and Intermediate Control of a Graphic Display." *Tech. Report 221 Series 2*. Department of Statistics, Princeton University, Princeton, New Jersey.

Tukey, J.W. 1983. "Another Look at the Future." In *Computer Science and Statistics: Proceedings of the 14th Symposium on the*

Interface. Troy, New York. Springer-Verlag, New York, New York, pp. 2-8.

Thomas, J.J. 1983. "A User Interaction Model for Manipulation of Large Data Sets." In Computer Science and Statistics: Proceedings of the 14th Symposium on the Interface. Troy, New York. Springer-Verlag, New York, New York, pp. 118-128.

LINKING SCIENCE, TECHNOLOGY, AND ECONOMICS DATA

Francis Narin

CHI Research/Computer Horizons, Inc.

Cherry Hill, New Jersey

LINKING SCIENCE, TECHNOLOGY, AND ECONOMICS DATA

Francis Narin

CHI Research/Computer Horizons, Inc.

1. Introduction

In this paper we will attempt to summarize some of the state-of-the-art of linking science, technology, and economics data. We wish to say at the outset that Computer Horizons, Inc. (CHI) specializes in science and technology indicators development, based on literature and patent citation analysis. For example, we analyze the linkage between technology and science by analyzing the citations (references) from patents to scientific papers.

As science and technology analysts, CHI is very much involved in database management problems — in the sense that we are constantly attempting to relate data from large databases — such as the *Science Citation Index* — and our own patent citation files — with other data. For example, we have been active in linking publications with universities, patents with companies, publications with patents, and funding data with institutions.

All of these linkages are difficult. Within the *Science Citation Index*, especially in some of the older years, there are as many as 300 different variants of the name "Harvard University" corresponding to different departments, different sections, and affiliated institutions such as hospitals, which may have different names, and so forth.

Even within the patent files there are sometimes dozens of variants of a company's name stemming from mechanical differences in abbreviation from one year to another, changes of name, errors in input, and so forth.

Even with funding data in government files there are difficulties in identifying what is meant by a given university. Sometimes campuses are handled separately, and sometimes they are combined, and one has to be very careful to know the vagaries of each individual data system.

As another example, when dealing with a German scientific publication in recent years, one has to be very careful to split East and West Germany. This has lead to our having, in the office, maps of Berlin with a line dividing East and West Berlin, and various institutions located on a block-by-block basis, so that we know whether the publication should, in fact, be attributed to the Federal Republic of Germany or the German Democratic Republic.

Further, even linking two scientific publications databases is not easy. The abbreviations for journals in the National Library of Medicine's MEDLINE system are totally different from those in the Science Citation Index: further, journals split, they combine, they change their names, abbreviations are changed on a year-to-year basis, and it is no simple task to, for example, know which journals are in common amongst the SCI, with some 3,000 journals, the MEDLINE with an equal or larger number

As a final correlate to that, just the way special issues, technical appendices, and so forth, are handled from one data system to another adds difficulties to making the linkages.

With all those negative comments in mind, we will now describe a substantial body of knowledge that shows that — despite the noisiness and mechanical difficulties of dealing with the data — there is, in fact, a very high degree of correlation between economic, scientific and technological data. We just warn that one has to be very very careful in doing these linkages, and that it is often very tedious to get the data into useable form.

2. Discussion of the Data

2.1 International Data

In this section of the manuscript we are going to briefly annotate the series of figures that will be used at the presentation. To keep this manuscript within reasonable size, since there are a larger number of figures, the annotations will be relatively brief. Much of this material is drawn from our reports "Evaluative Bibliometrics: The Use of Publication and Citation Analysis in the Evaluation of Scientific Activity" and "Subjective vs. Bibliometric Assessment of Biomedical Research Publications". See the Bibliography for full reference on these and other sources.

Figure 1 shows a histogram of the number of publications in the field of animal anatomy published between the year 1550 and 1860, adapted from a paper published in 1917 by Cole and Eales. This data is the oldest quantitative data of which we are aware on scientific publication. The Cole and Eales paper is a beautifully written document, discussing everything from the problem of which country to attribute a publication to the country of birth or education of the author, or the country in which he is working — or in those days, the country in which it was published because of ease of publication — to the effects of war and birth rate on science in Europe.

Of particular note is their comment that, in the first hundred years of anatomy publication the pattern was sporadic because there were just a few books and scattered investigators — there were no organized colleges and groups, per se, to pass the knowledge on to.

With the founding of the Royal Society, and of the first scientific journals in the 1660s, the publication pattern became somewhat more stable, but still suffered over the next 150 years from being individual rather than institutional. There were still no long lived groups of animal anatomy researchers, but rather individuals who sometimes left students and followers, and sometimes didn't.

Cole and Eales make a particular point of the emergence of major growth in 1815, at which time two things happened: first, the solid emergence of serial publications — 100 or more journals were in existence by then — and second, the end of the Napoleonic Wars and the settling of peace in Europe. These two events truly started the first major growth in anatomy publications, to a peak of a few hundred per year in the 1830s.

Thus we see that it was known, some 70 years ago, that scientific publication is dependent upon external, historical factors. This is shown somewhat more specifically in Figure 2, which plots GNP versus the number of abstracts in Chemical Abstracts for the U.S.A. over a relatively long period of time.

The main point is that the number of abstracts and the GNP are increasing at essentially the same rate. External events and science are not at all independent of one another.

Figu^re 3 takes this data a bit farther, and plots the number of Chemical Abstracts entries per dollar of GNP for six major countries over the early 1970s. It is interesting to see how closely five of the countries, the U.S.A., France, Italy, Germany and Japan converge while the U.K. is quite different by 1973. Our interpretation of this is simply that while the economy of the U.K. was slipping badly, the U.K. publication rate stayed at its traditional level -- which may be related to a paper by Langrish which showed that, within U.K. chemistry, university papers and industrial papers are very highly disconnected. The implicit relationship between industrial productivity and chemical papers, which exist in Figures 2 and 3, may be unusually weak in the U.K., where there is a certain separation and disdain between university and industrial chemists.

A side note: in this data, which is our own, we attempted to add Soviet GNP and Soviet origin abstracts. We encountered difficulty in establishing any kind of comparable figures for Soviet GNP. We would argue, however, that it may well be that one interesting way of getting an idea of the size of the Soviet industrial complex would be to look at Soviet publication and patent data.

Along those lines, Figure 4 shows number of biomedical papers versus 1972 GNP for a large number of countries. By taking the log of both sides, one gets a correlation of 0.85 between GNP and publications, and most countries lie fairly close to the regression line.

This data is based on the Science Citation Index which has some particular biases in its coverage, especially against non-English language papers, and papers in Cyrillic or other languages from which it is difficult to translate. Thus, it is noticeable on the curve that the number of Soviet Union biomedical publications is low, as are those of Japan and Poland.

The countries that are particularly high in biomedical publication versus GNP tend to be those with strong social democratic governments, the U.K., Canada, Sweden, Australia and Israel. Switzerland is high, perhaps, because of the concentration of pharmaceutical companies in Switzerland.

2.2 Programmatic Data

Figures 5 and 6 take the same type of data down to a more precise level of disaggregation, and look at the funding for Bureaus, Institutes and Research Divisions (BIDs) at NIH, and compare this with the number of papers which acknowledged NIH research support.

To generate this data, which is part of some ongoing work at CHI, we manually looked up the research support acknowledgements in some 800,000 papers published between 1970 and 1980. Fortunately, in 1981, the National Library of Medicine started to add this data to its database, so as we extend this data in the future, we will not have to look up these papers.

It is particularly clear, in comparing the two figures, that funding and publications correlate highly. Looking at 1977, for example, the order of the top four institutes is identical on the two figures; within the next three institutes there is a slight permutation; the two smallest institutes, the Eye and the Dental Institute, are together at the lower parts of the curves.

Even the minor differences between the institutes are easily explicable. The number of papers from the National Institute of General Medical Science (NIGMS) is slightly higher than the National Institute of Arthritis Metabolic and Digestive Diseases (NIAMDD), even though

NIGMS funds are a bit lower. However, NIGMS is the basic research institute at NIH, and would be expected to have more papers per funding dollar, simply because there is more lab type research supported by NIGMS and less of the more expensive, clinical type research.

A most noticeable aspect of those two figures, of course, is the enormous increase in funds for cancer research, which occurred following the National Cancer Act of 1970, and the acceleration in cancer publication which followed two or three years later. Similarly, the National Heart Long and Blood Institute received an acceleration in funds with the Heart Lung and Blood Act in the early 1970s, which is reflected in its increase in publications also. Funds and publications, at the support source level, do correlate closely.

That this correlation continues down to the institutional level is illustrated by Figure 7, which shows the total number of papers coming from 110 U.S. medical schools, essentially all significant medical schools in the early 1970s, versus the amount of funds the schools receive from NIH. The correlation on that curve between funds and publications is 0.95, and the only school that is notably off the curve, Harvard, is easily explained. Harvard Medical School has a substantial endowment, and receives an unusually large amount of funding from sources other than the federal government: both factors tend to raise the number of Harvard Medical School publications per NIH dollar.

This kind of data can be taken even further to analyze institutional characteristics. Figure 8 shows the Gini Index for dispersion of papers in medical subfields versus medical school research (publication) size. The Gini Index is a measure of concentration. If all papers from a medical school were in one biomedical subfield, then the Gini Index for that school would be 1.0. If the papers from a school were uniformly distributed across all possible biomedical subfields, then the Gini Index would be 0.0. Thus, the steady decrease in Gini with the medical school size simply says that, as schools get bigger, they publish papers in more and more subfields. This is not exactly a startling conclusion, but one that again shows that publication and institutional characteristics relate quite extensively.

Figure 9 takes another view of institutional data, and looks specifically at the relative emphasis of one particular institute at NIH, the National Institute of Allergy and Infectious Diseases, on different subfields of science. If the publications resulting from NIAID support were equally distributed across all subfields of basic biomedical research, in proportion to the size of the subfields themselves, then all activity indices would be one. The fact that for three subfields, microbiology, virology, and parasitology the activity indices are in the range of 3, 5, and 8-9, indicates a high degree of concentration of NIAID activity in those three subfields. Those three subfields of basic research are obviously quite closely related to allergy and infectious diseases. Other institutes at NIH show entirely different activity profiles. In this case, what we are seeing is that the congressionally assigned mission of the institute at NIH is clearly reflected in its publications, which are quite heavily concentrated in basic subfields related to the research mission of the institute.

Figure 10 deals with another aspect of scientific activity, that of the geographic correlates. Scientific publication, like most human activity, is extremely skewed in its distribution, being concentrated in specific cites and areas. Figure 10 shows this for Canadian science using a measure of scientific potential -- authors per kilometer where, to some degree, the potential is defined as the sum of the number of authors divided by the distance between them. Particularly apparent on the figure are the concentrations around Ottawa and Montreal. Various other aspects of this, from the same paper, show that Canadian science is concentrated even more heavily than Canadian population: there are clear correlates with governmental and university location above and beyond population concentration.

2.3 Linkage Within Science

In the next few figures we will turn our attention to some of the happenings within science, rather than correlates with the outside.

Specifically, Figure 11 is what we call a two-step model of chemistry journals, for 1969. Specifically, for each journal we drew two arrows to the journals which the original journal cites first and second most frequently. Even a glance at the figure shows that the literature of chemistry is highly organized. The Journal of the American Chemical Society is obviously the key, central journal. We also have its British equivalent, the Journal of the Chemical Society A. B. C. in the left center as another focus and the Journal of Chemical Physics, in fact, cites most frequently the Physical Review, and that journal is a linkage point between chemistry and physics.

There is also another cluster of journals around analytical chemistry shown in the upper right.

Thus, one could conclude here that, again, science does not exist as an amorphous mass, but that there is a substantial structure within a scientific field.

That this can be generalized from one scientific field to another is shown in Figure 12 from that same paper. This is a two-step cross-field model. In this model all the journals within a field are represented by rectangles and journals which are cited first or second most frequently by journals in more than one field are shown in ovals.

Thus, from the previous figure, almost all the chemistry journals are in the one central box for chemistry. The Journal of Chemical Physics, which is cited first or second most frequently by a host of chemistry and physics journals, is shown separately as are a few other journals which seem to lie on the boundary between chemistry and physics. Appropriately enough, the Journal of Fluid Mechanics lies at a boundary between physics and mathematics-statistics, and a group of biochemical journals lie in the areas between biology and biochemistry as a field, and biochemistry and chemistry as a field.

The two special journals Science and Nature are multidisciplinary journals, and link very directly to physics and mathematics as well as to biology.

Figure 13 is another view of the linkages between scientific fields, and shows the specific citations from journals in the subfield of cancer to journals in other biomedical subfields. In this case, each subfield is defined as a group of journals, and the references from one cancer journal to another are not shown. Note that cancer journals cite heavily to biochemistry, immunology, virology, cell biology, and some of the other biomedical fields. They also cite rather heavily to three special journals, Proceedings of the National Academy of the U.S., Nature, and Science which are very general journals in biomedical research.

The background of that map is, by the way, a mapping of biomedical subfields in accordance with (on the X axis) the ratio

$$\frac{\text{references to biochemistry}}{\text{references to physiology}}$$

and on the Y axis the influence weight — a measure of how frequently the subfields are cited. Thus, the more heavily cited journals or subfields tend to be toward the bottom (of higher citation influence) and the more biochemical tend to be on the left.

In fact, the ratio cites to biochemistry over cites to physiology is twenty-five times larger for papers in virology journals than it is for papers in respiratory diseases: nothing is linearly distributed in science!

Figure 14 shows how rankings of university departments correlate with various bibliometric measures. In this case the Roose-Andersen ranking is a ranking of U.S. departments of physics, biochemistry, chemistry, etc. against one another, done by survey of thousands of university faculty members. We compared these peer rankings of the departments with rankings of the departments obtained from our bibliometric data. Those basic comparisons are shown on the diagonal line. For example, a publication ranking (number of papers published by the department) versus a Roose-Andersen ranking of departments in physics correlates in excess of 0.9; biochemistry and chemistry department rankings correlate with numbers of papers in the 0.8 range — psychology, microbiology, mathematics, physiology and in the 0.7 range — while pharmacology, zoology and developmental biology correlate between 0.6 and 0.7.

When the ranking of a department is based both on the citation frequency of the department's papers and on the number of these papers, then in every single case the agreement between the peer ranking and the bibliometric ranking increases. In psychology and in mathematics the increase is particularly large, but in all the fields the comparison of peer perception and bibliometric perceptions of the universities are highly correlated. In fact, the best bibliometric model of the Roose-Andersen rankings is the sum of two terms; one is the publication size of the department and the other is its citation rating.

In that study we also found that there is a considerable halo effect in peer rankings: within a single university there is a much higher correlation of the ranking of the departments when the rankings are based on peer perception than when based on bibliometrics. The fact that the University of Chicago has such strong departments of physics and mathematics appears to wash over into perceptions of the strength of other departments in the university.

2.4 Productivity

Many of the quantitative studies of scientific productivity are reviewed in Chapter IV of our monograph "Evaluative Bibliometrics". The reader is referred to that for further details. We will touch on just a few of the high points in the next few figures. These deal very briefly with some of the questions of productivity in science — its fundamental laws, and how it may be compared to peer perceptions.

Specifically, Figure 15 illustrates a very fundamental law of bibliometrics "Lotka's Law" first promulgated in 1926 for Chemical

Abstracts, and since verified many times in many different fields. The essence of that law is that scientific productivity goes by a $1/N^2$ law.

If you have 100 authors who produce one paper in the course of three years, you will have $100/2^2$ or 25 who will produce two papers, $100/3^2$, or 11 who will produce three papers, and so forth. This is essentially the bibliometric equivalent of the famous 20/80 rule of the thumb, which says that a relatively small proportion of people produce most of the work, in any situation.

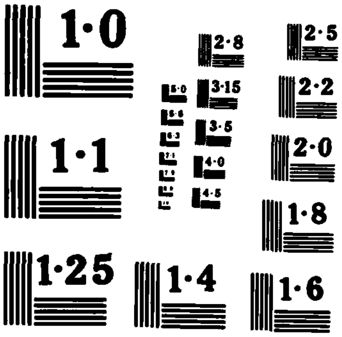
There is further evidence of an even steeper distribution of highly cited papers — in some fields approaching $1/N^3$ or more. This type of law has also been shown by Schockley to hold for the distribution of productivity within a large laboratory, and certainly seems to be a fundamental characteristic of science, and, as we will show later, technology also.

Another aspect of scientific productivity, is shown somewhat incorrectly in Figure 16, which is a plot of age versus contribution to chemistry from a study of Lehman's. The flaw is, this data does not take into account the large number of chemists who leave the bench at a relatively early age as they become administrators, or teachers, and take on other duties peripheral to research. There are studies that show that, for people who stay in research, their productivity is essentially flat into their sixties.

2.5 Technology and Science

In the remaining parts of this paper we are going to discuss some data from our patent database, and then some linkages between patent data and literature data.

AD-A158 285 DATA BASE MANAGEMENT: PROCEEDINGS OF A CONFERENCE 3/4
NOVEMBER 1-2 1984 HELD AT MONTEREY CALIFORNIA(U)
STANFORD UNIV CA DEPT OF STATISTICS H SOLOMON ET AL.
UNCLASSIFIED 31 JUL 85 N00014-76-C-0475 F/G 9/2 NL



NATIONAL BUREAU OF STANDARDS
MICROCOPY RESOLUTION TEST CHART

Most of this data is based on a relatively large patent citation file that we have in-house — containing data on the 5,000,000 citations on the front pages of 800,000 U.S. patents issued since 1971.

The first figures in this section, Figure 17, shows that the foreign dependence of U.S. patents is increasing in scientific instrumentation. This is typical of virtually all fields in the patent system. In fact, throughout the U.S. patent system itself, the total number of U.S. patents of foreign origin, e.g., foreign invented and, or foreign owned, is now close to 50% with the Japanese share close to 15% of all U.S. patents. Thus the U.S. patent database is very international, and U.S. patents may be quite applicable to the study of foreign technology. Although the database is clearly biased towards U.S. origin inventions, a study by Soete and Wyatt reveals that it is relatively unbiased in its representation of foreign technology. Thus U.S. patent activity may be a rather good way of comparing British, German, French, and Japanese technology. Since there are some 70,000 U.S. patents per year, and half of these are of foreign origin, the amount of data available for U.S. — foreign comparisons is quite substantial.

In any case, Figure 17 shows that foreign origin — U.S. patents are much more heavily dependent on foreign origin material than U.S. origin — U.S. patents; that is, foreign origin U.S. patents cite foreign patent systems roughly twice as frequently as U.S. origin U.S. patents, and foreign origin U.S. patents cite other foreign origin U.S. patents about twice as frequently as U.S. origin U.S. patents. One interesting implication is that this indicates a substantial flow of technological and scientific knowledge from foreign sources, rather than the reverse. In fact, some further analysis we did on multinational corporations seemed to indicate that even the U.S. invented patents of foreign multinationals show a much stronger dependence on foreign sources than one would expect for U.S. origin patents, e.g., it may be that the foreign multinational corporations operating in the U.S. do, in fact, bring in technological knowledge, rather than the reverse.

One important point about Figure 17 is that, for the figure, we assigned patents to scientific instrumentation through the use of an SIC-to-patent class concordance which has been developed by the Office of Technology Assessment and Forecast, of the U.S. Patent Office, specifically, for this kind of assessment. The concordance, in essence, assigns a given patent subclass to an SIC code. The concordance is one approach to a fundamental problem with the U.S. patent classification system. The U.S. system, is, in essence, an art based classification: a jet engine turbine, a fan and a windmill are all rotating devices and are liable to be classified far more closely than their application or industry would require. Therefore, it is an enormously difficult task to relate patent statistics to financial statistics, which are almost entirely in SIC category for the U.S. Furthermore, attempts to do this, by going to the company level, and using company patents and company economic data, run into the problem that detailed financial data is often not available for companies. For example, it is often impossible to separate the financial data of, for example, the bus and the car parts of an automotive company, even though patents related to buses and automobiles might be easily separable in a patent database. Linking technology to economic data at a detailed level is not a trivial problem.

Since this is a database meeting, we have to also call attention to the fact that there are substantial difficulties in dealing with company identification in a patent database. Companies have many different names in the patent database, some of them minor variations (Exxon, Esso, etc.) some of them typographical errors, some of them changes in the way data are entered from year-to-year, and so forth. Each year, as we update our patent data, we take approximately a man-month to identify the thousands of new company names, and to see if they are the same or different companies from the names that have already been identified.

Relating this data to other company data — for example, publication data, sales data, and so forth — is also a non-trivial task, both because of variations in name, but also because of the seething merger acquisition of market in the United States. A company may be a subsidiary one year, independent the next two years, and a subsidiary of somebody else four year's down the line. Thus, each time we look at company performance, we have to essentially decide what we mean by the company first.

Not unrelated to that you may recall, a year or two back, that there was a large amount of publicity on the various merger maneuverings of Martin, Bendix, and Allied Chemical. There was also, of course, the prosecution of Hitachi for stealing IBM manuals and other information.

While all that was going on we decided to attempt to use our patent data to ascertain whether companies are similar, in terms of their patenting. We defined an "index of similarity" based on the concentration of patents in the major U.S. Patent Office classes. It was possible to see from Figure 18 that there really is very little technological resemblance between Martin, Bendix and Allied. The largest index of similarity is 0.32, between Martin, and Bendix.

When we looked at Hitachi, IBM, and Burroughs we expected to find much higher correlations and, in fact, we did, but not as high as expected. In fact, the Hitachi patent pattern correlates with IBM and Burroughs only on the range 0.5-0.7 somewhat to our surprise. However, when we looked at the patents themselves, we found that Hitachi owned an elevator company, and a large number of patents in the elevator area are assigned to Hitachi. When we took those patents out, then the correlations between Hitachi patents and IBM and Burroughs were much closer to the relatively high correlation of 0.87 between IBM and Burroughs. Technologically similar companies do patent in technologically similar areas: you just have to be very careful to compare those aspects which are truly comparable.

Another way of looking at technological relationships between related companies is shown in Figure 19. Because this analysis was done for a private client, the company identification is coded. For this figure we used citation data to show that a parent company and its two subsidiaries, Subsidiary 1 and Subsidiary 2 are not very highly linked. The parent company, at the bottom, cites and is cited mainly by companies other than its two subsidiaries. In fact, at a level of five citations, the parent and Subsidiary 2 are not linked at all, and the parent and Subsidiary 1 are barely linked. On the left "I," which totally coincidentally happens to be IBM, has 16 patents which cite Subsidiary 1's patents and 17 patents which cite the parents patents. Similarly on the right, IBM's patents are cited 15 times by Subsidiary 1 and 21 times by the parent. Thus, IBM is a "link" between the parent and Subsidiary 1. However, IBM is one of the most frequently patenting companies in the U.S. patent system. This link is probably more characteristic of the size of IBM than of any strong technological linkage between the parent and its subsidiary.

We might add, in passing, that the acquisition of these two subsidiaries has not been a very lucrative one for the parent company, which is an old line mechanical company which attempted to diversify into modern technology by these acquisitions. Our point is that these patent mapping techniques reveal that the parent and its subsidiaries are quite technologically separate.

The last few figures in this discussion deal with linkages between patents and science, and make the point that these linkages are perhaps far stronger than people expect, and that they are certainly far more current than people expect. We find that high technology patents cite heavily to the very recent scientific literature across the entire spectrum of basic to applied research.

Figure 20 summarizes the number of citations on the front pages of U.S. patents, and shows that citations to non-patent sources are roughly half to scientific journal articles, and half to various other categories. The categories not shown are widely scattered across all the rest of published material.

Figure 21 shows the distribution of 6,500 references contained in 399 prostaglandin patents. When we did this study, prostaglandins was one of the most actively growing areas of U.S. patents. The main point of that figure is that more than half the references in prostaglandin patents are to papers in scientific journals. Furthermore, the majority of the cited papers are in journals covered by the Science Citation

Index — the central core of the scientific literature. This shows a far greater linkage to the scientific literature than we really expected to find in a technological source such as patents, and indicates that much of modern high technology is virtually indistinguishable from science. Many of these prostaglandin patents read exactly like standard scientific papers. The order of the discussion is prescribed somewhat more strictly in patents, but the level of scientific competence, and the manner in which those patents draw upon the reservoir of scientific knowledge is virtually identical to any first class scientific paper.

With the fact that patents link to science in research, Figure 22 looks at the time distribution of the linkage. In the figure we compare

patents citing papers,

patents citing patents,

papers citing papers.

The main point is that both the papers and the patents are citing material that is relatively recent. The peak in citation: patents to papers, and patents to patents, is three to five years; the peak in papers citing papers is two to three years. If we allow an extra year or two for the prosecution of the patent, then clearly the patents are using technology and science which is just about as recent as the science used by science. Technology today links to today's science! It is not linking to the science that has been categorized and codified and put in text books to be read by engineers in school. It is linking to the science that is being published on a daily basis. In some areas of biotechnology it may be that the time lag between publication and patenting is measured in weeks rather than in years.

Another interesting aspect of the similarity between science and technology is shown if Figure 23, which plots the citation distributions (number of patents and papers cited n times) for patents and papers in the biological area. The main point — shown by the dashed curve — is that both those distributions are skewed, the distribution of cited patents being much more heavily skewed than the distribution of cited biomedical papers. In science, in general, the distribution of highly cited papers is somewhat steeper than the $1/n^{1.5}$ shown. However, this data is from the core biomedical journals, and the papers are, therefore, more highly cited than all papers. Further, biomedical papers are more highly cited than papers in other fields. Nevertheless, there is certainly a skewed and non-linear distribution for cites to biomedical papers, and an even more skewed distribution of cites to biotechnology patents. If one makes the assumption, as is often made, that highly cited papers represent the small number of key events which drive science, one could argue that the bibliometrics of patents indicates that the same thing is true in technology: technology, as well as science, may be driven by a relatively small number of important events.

The last data figure, Figure 24, shows the types of papers that are cited by seven rapidly growing subclasses of patents. This figure categorized the cited scientific papers in terms of research level, from Level 4, basic research journals, through to Level 1 and 2, engineering and technology journals.

The main point of this is that, essentially, the patents are citing very evenly across the basic to applied research spectrum. Approximately one third of science is in Level 4 journals, one third in Level 3 journals, and one third in Level 1 and 2 journals. There does not seem to be any preferential citing from patents to, for example, the more applied journals. Patents are using all of science, from basic to applied, essentially in proportion to the number of papers in each of the levels. Thus we maintain that the linkage from technology to science is, essentially, from technology to all of science: it is not to any one particular aspect.

3. Conclusions

The major conclusions of this paper can be stated succinctly:

1. Funds and scientific publication are tightly linked at national and institutional levels.
2. Cross-national, cross-institutional, geographic and subject-to-subject science linkages are quantifiable and mappable, and reveal that the spectrum of activity from funds, to pubs, to patents, to productivity is reflected in highly correlated statistics.
3. Citations from patents to papers document a close day-to-day relationship between science and technology.
4. The science used by patented technology is very recent — implying little or no lag between discovery and utilization.
5. Technology like science may be driven by a small number of key highly cited events.

4. Bibliography

Anderson, Richard C., Narin, Francis and McAllister, Paul R. "Publication Ratings vs. Peer Ratings of Universities", Journal of the American Society for Information Science, 29, 2, 91-103, March, 1978.

Carpenter, Mark P., Cooper, Martin and Narin, Francis. "Linkage between Basic Research Literature and Patents", Research Management, 13, 2 30-35, March, 1980.

Cole, F.J. and Eales, N.B. "The History of Comparative Anatomy", Science Progress XI, 578-596, 1917.

Frame, J. Davidson and Narin, Francis. "International Distribution of Biomedical Publications", Federation Proceedings, 36, 6, 1790-1795, May, 1977.

Inhaber, H. and Prezednowek, K. "Distribution of Canadian Science", Geoforum, 19, 45-54, 1954.

Langrish, J. "The Changing Relationship between Science and Technology", Nature, 250, 614-616, August, 1974.

Lehman, Harvey C. "The Creative Production Rates of Present Versus Past Generations of Scientists", Journal of Gerontology, 17, 409-417, 1962.

McAllister, Paul R. and Narin, Francis. "Characterization of the Research Papers by U.S. Medical Schools", Journal of the American Society for Information Science, 34, 2 123-131, March, 1983.

Narin, Francis, Carpenter, Park P. and Berlt, Nancy C. "Interrelationships of Scientific Journals", Journal of the

American Society for Information Science, 23 5, 323-331, September-October, 1972.

Narin, Francis. "Evaluative Bibliometrics: The Use of Publication and Citation Analysis in the Evaluation of Scientific Activity", Contract NSF C-627, National Science Foundation. March 31, 1976. Monograph: 456pp. NTIS Accession #PB252339/AS.

Narin, Francis and Gee, Helen H. NIH Program Evaluation Reports: An Analysis of Research Publications Supported by NIH and DRR, 1970-1976: NIH and NCI, NIH and NHLBI, NIH and NEI, NIH and NIAID, NIH and NIAMDD, NIH and NIDR, NIH and NICHD, NIH and NIGMS, NIH and NINCDS, NIH and DRR, U.S. Department of Health and Human Services, Public Health Service, National Institutes of Health, December, 1980.

Narin, Francis. "Subjective vs. Bibliometric Assessment of Bicmedical Research Publications", NIH Program Evaluation Report. U.S. Department of Health and Human Services, Public Health Service, National Institutes of Health, April, 1983.

Roose, Kenneth D. and Andersen, Charles J. "A Rating of Graduate Programs", American Council on Education, 1970.

Shockley, William. "On the Statistics of Individual Variations of Productivity in Research Laboratories", Proceedings of the IRE, 279-290, March, 1957.

Soete, L.G. and Wyatt, Sally M.E. "The Use of Foreign Patenting as an Internationally Comparable Science and Technology Output Indicator", Scientometrics, 5, 31-54, 1983.

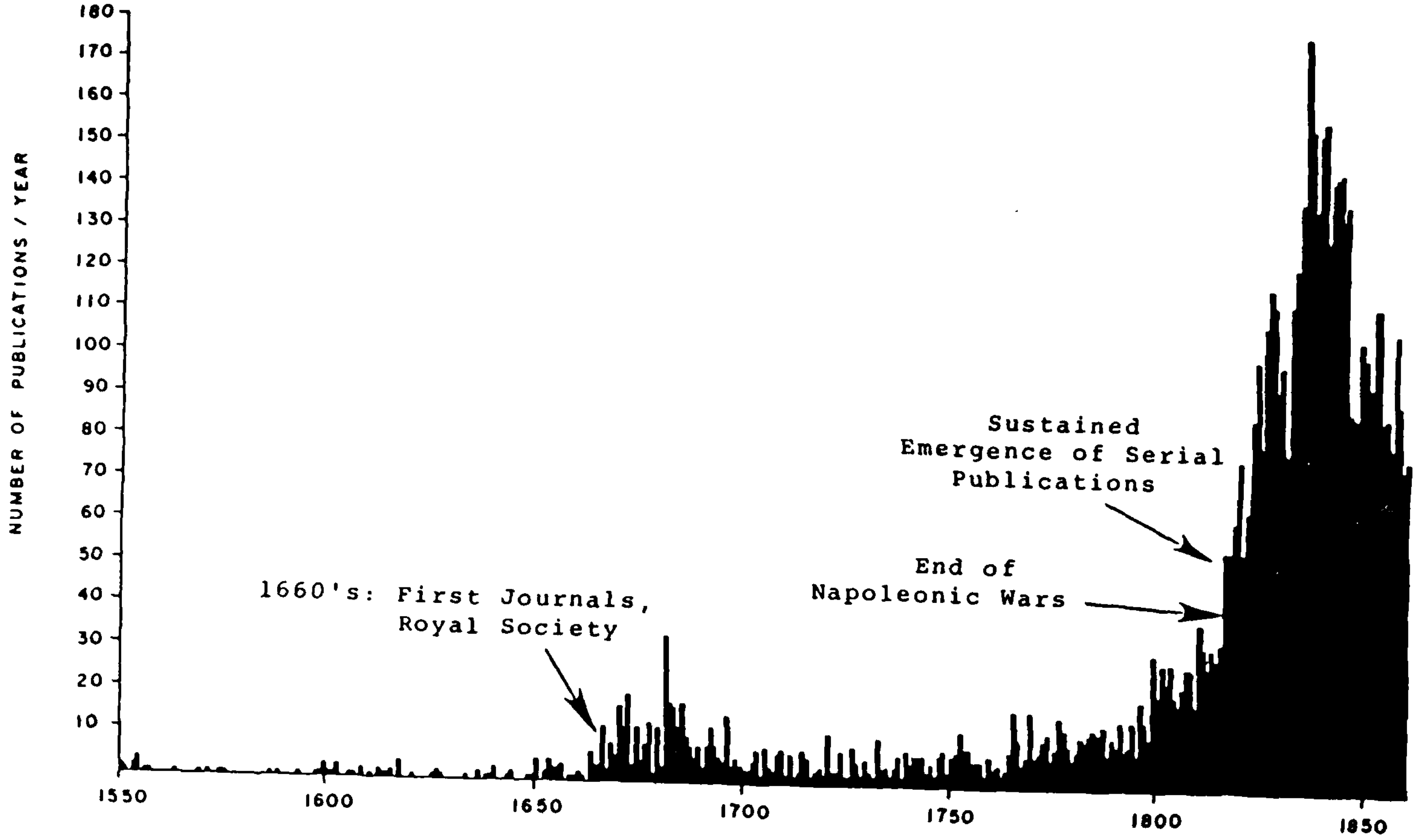


FIGURE 1

6434 COMPARATIVE ANATOMY PUBLICATION BETWEEN 1550 and 1860
(adapted from Cole & Eales, 1917)

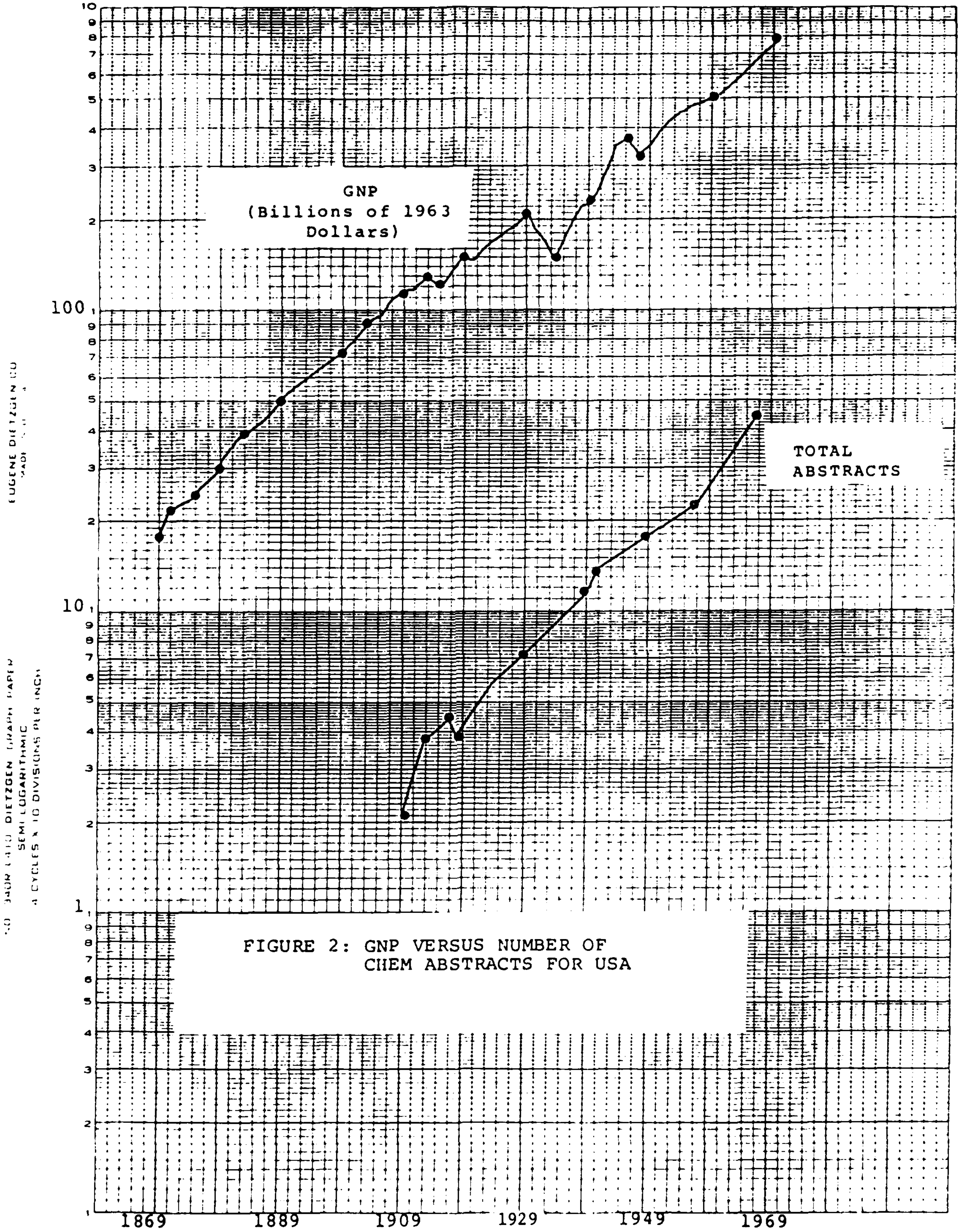


FIGURE 2: GNP VERSUS NUMBER OF CHEM ABSTRACTS FOR USA

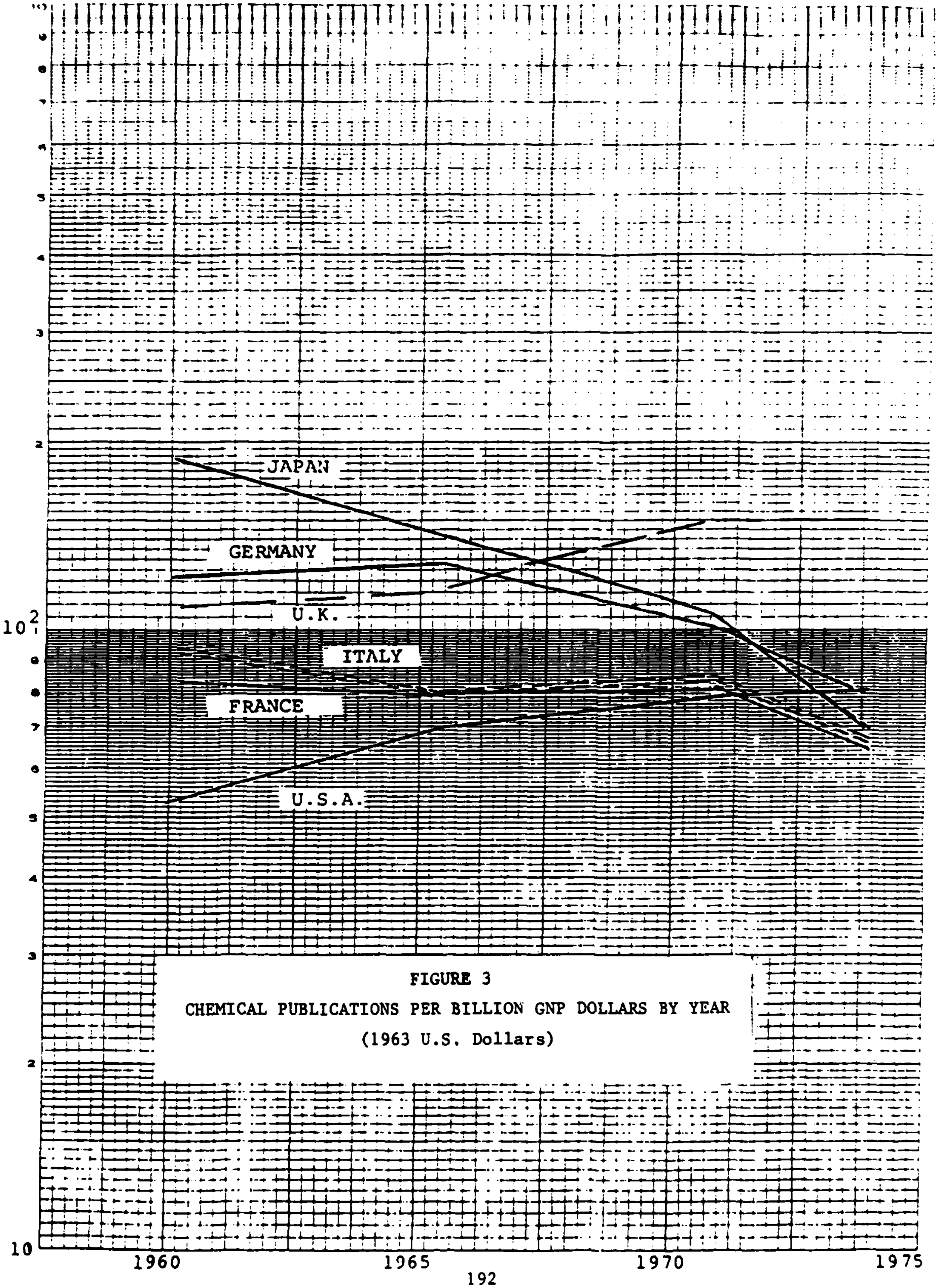


FIGURE 3

CHEMICAL PUBLICATIONS PER BILLION GNP DOLLARS BY YEAR

(1963 U.S. Dollars)

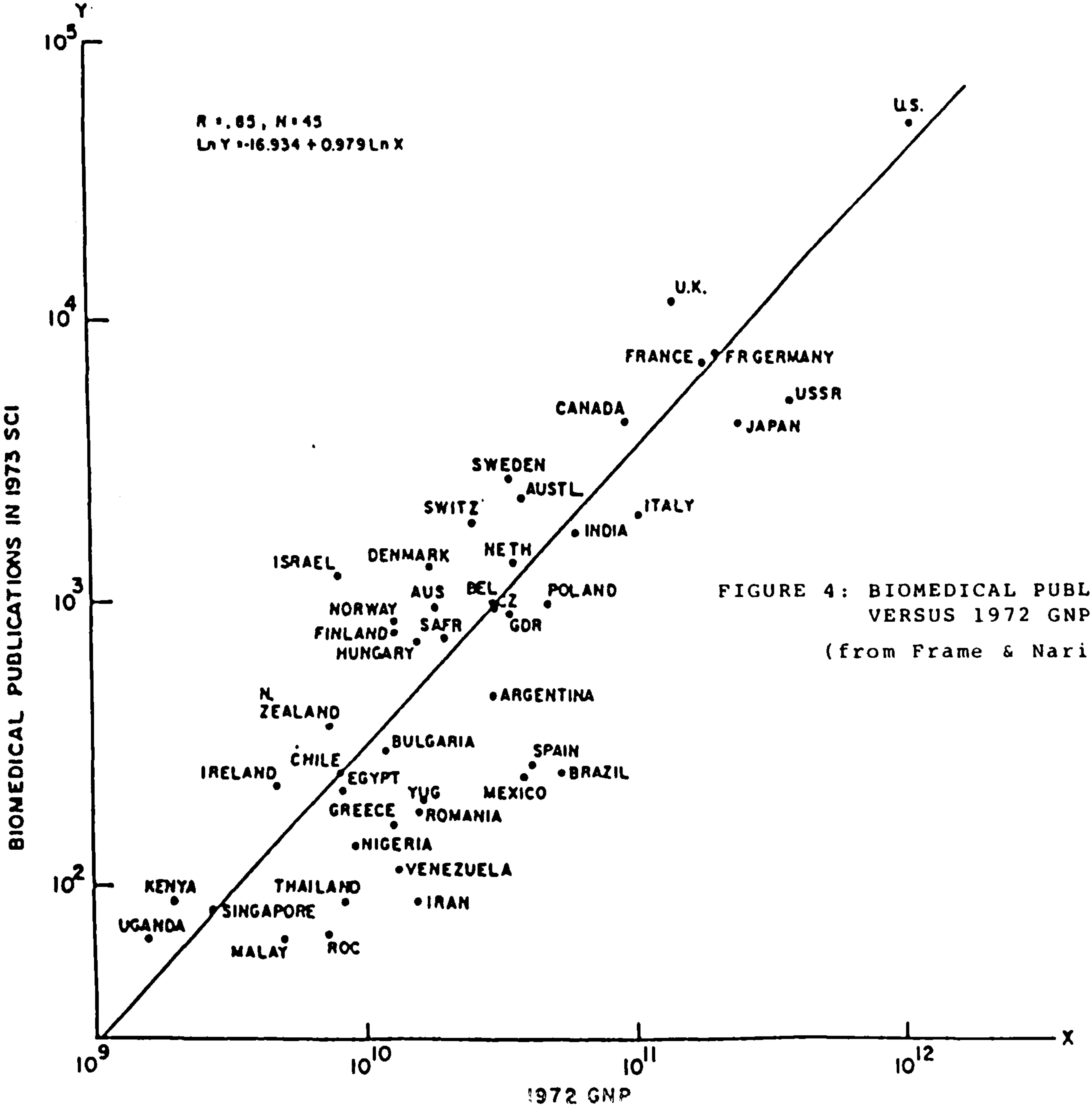
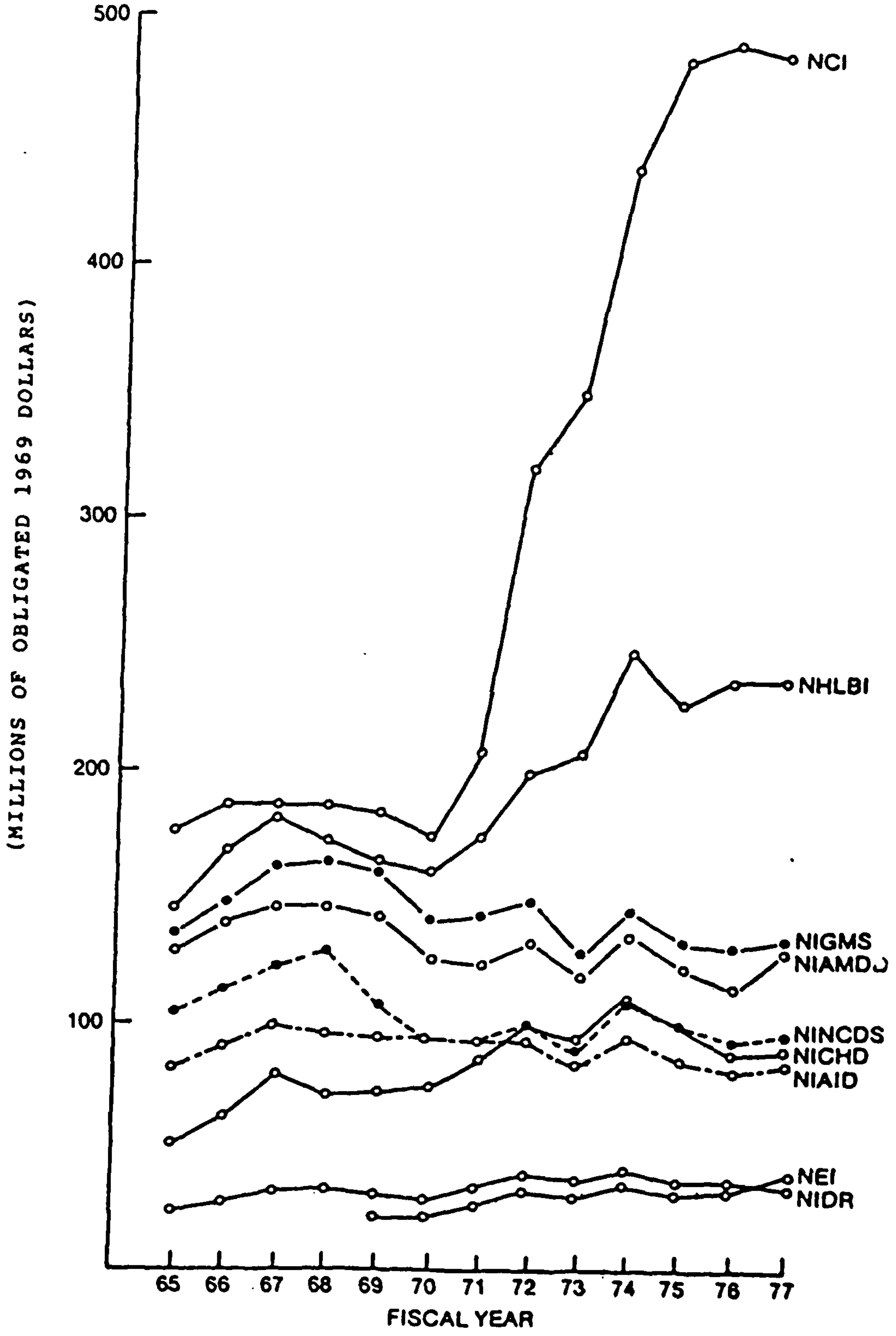


FIGURE 4: BIOMEDICAL PUBLICATIONS VERSUS 1972 GNP

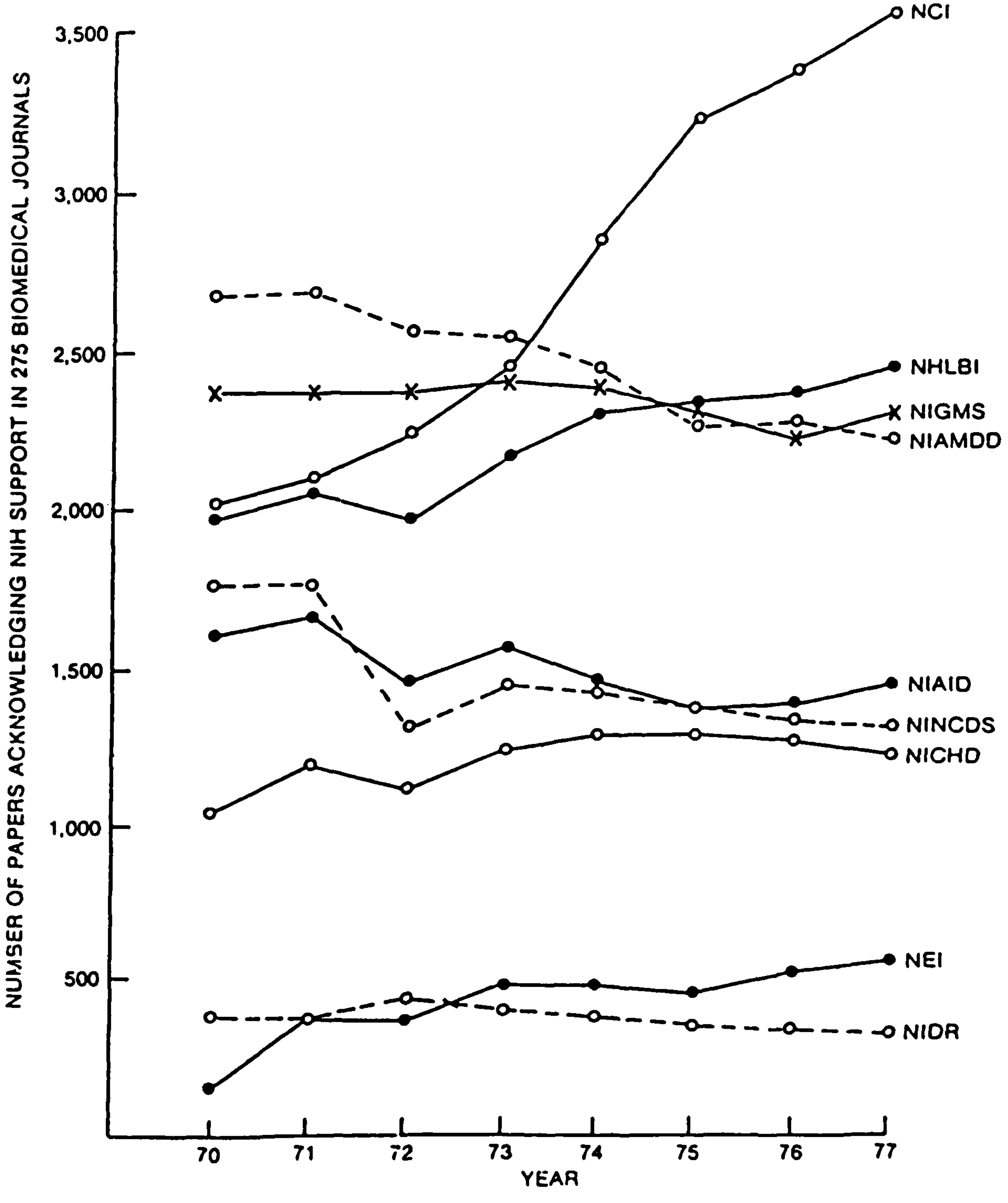
(from Frame & Narin, 1977)

FIGURE 5: FUNDING FOR THE BID'S



(from Narin, Gee, 1980)

FIGURE 6: ACKNOWLEDGEMENTS OF RESEARCH SUPPORT APPEARING IN 275 SCIENTIFIC JOURNALS



(from Narin, Gee, 1980)

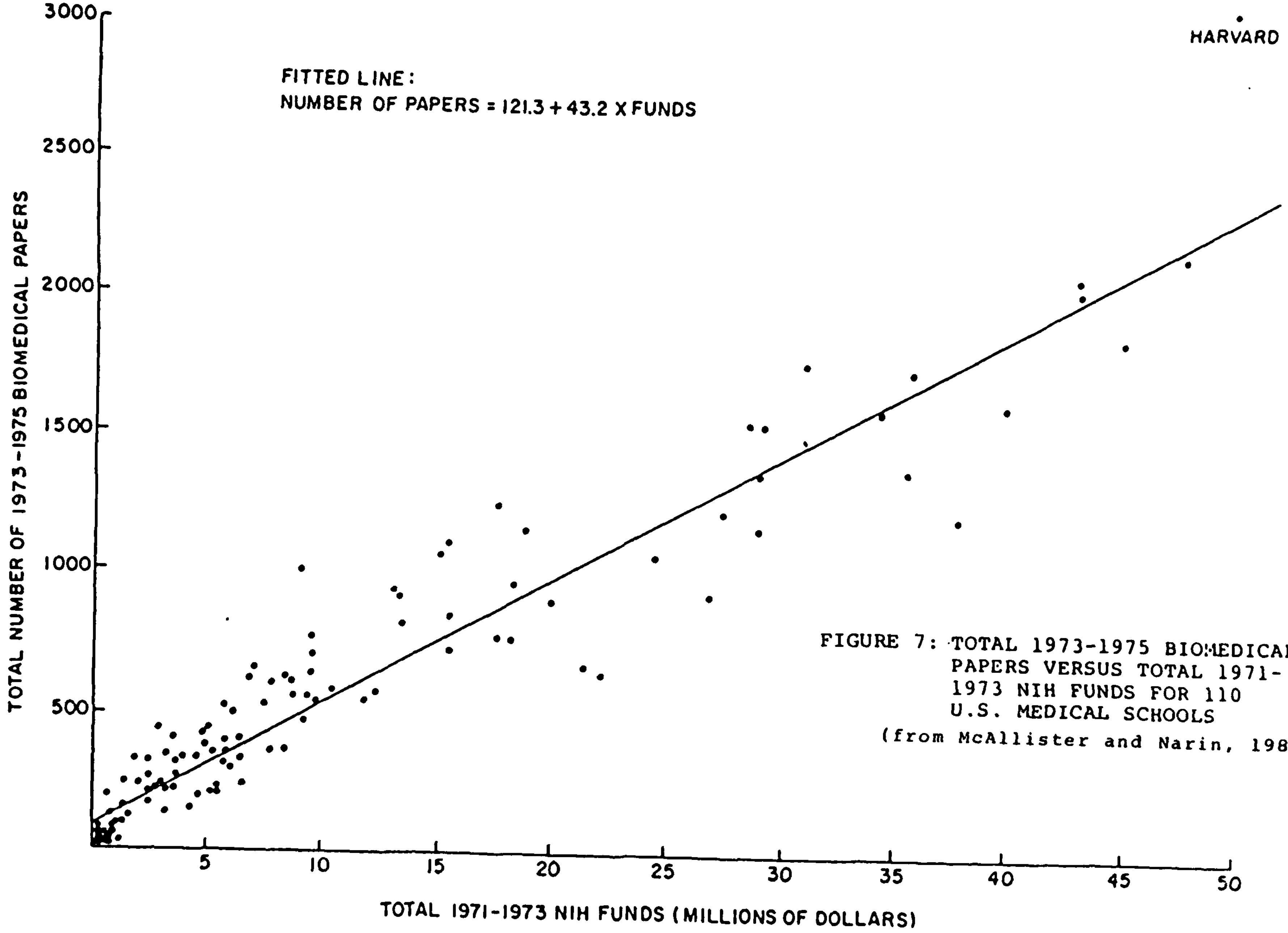


FIGURE 7: TOTAL 1973-1975 BIOMEDICAL PAPERS VERSUS TOTAL 1971-1973 NIH FUNDS FOR 110 U.S. MEDICAL SCHOOLS

(from McAllister and Narin, 1983)

FIGURE 8: GINI INDEX VERSUS MEDICAL SCHOOL RESEARCH SIZE

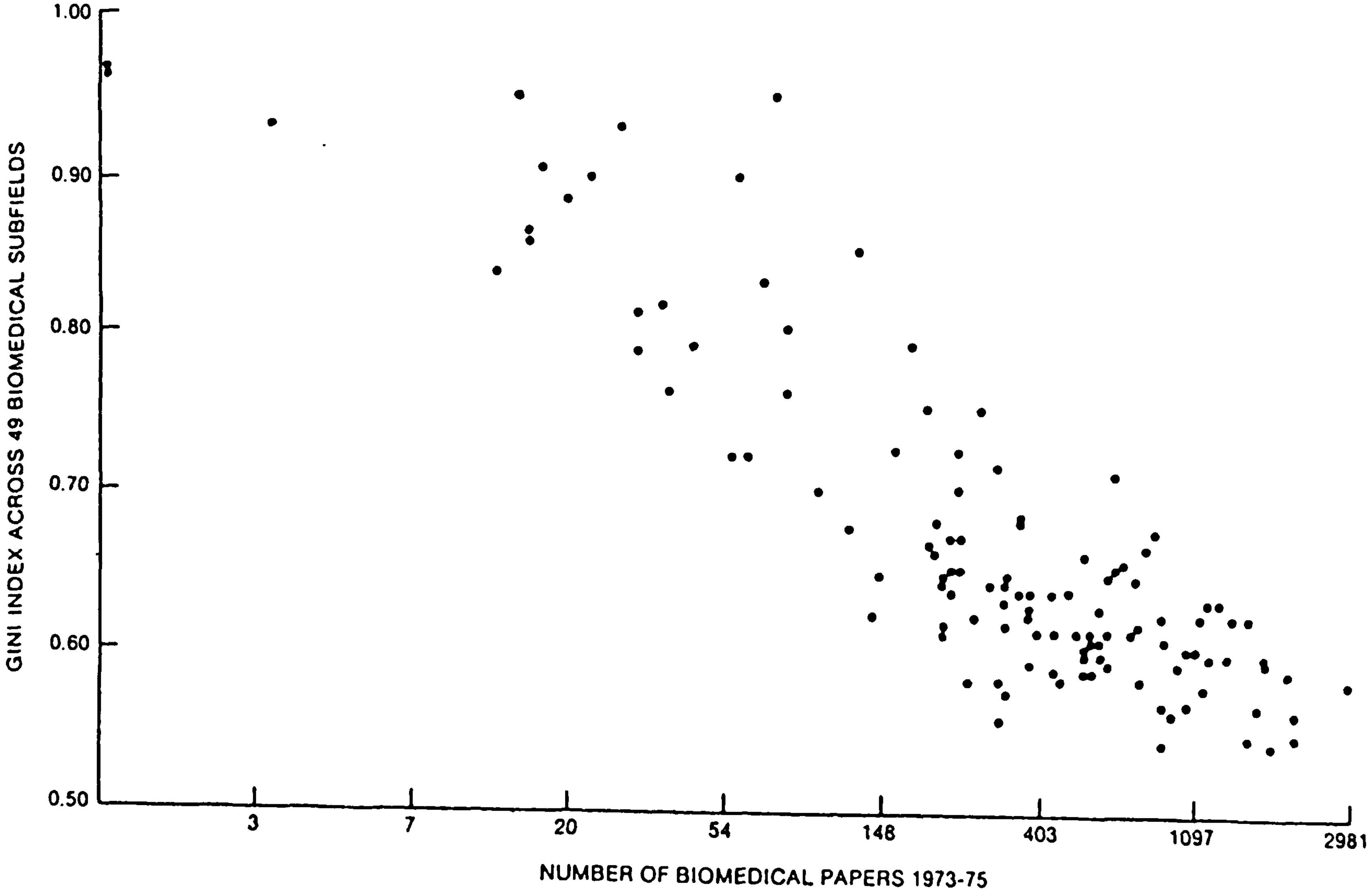
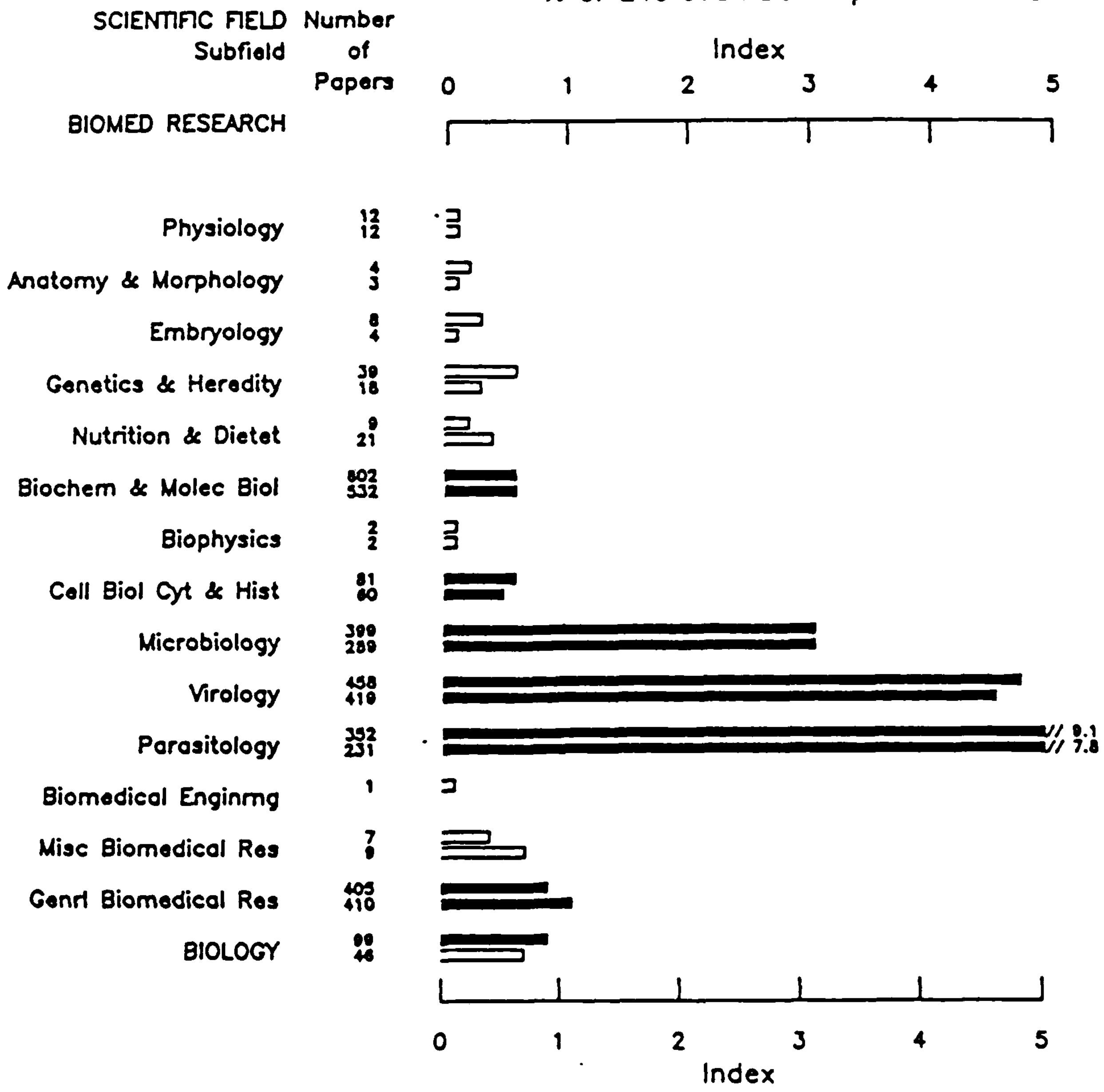


FIGURE 9

NIAID Activity Indexes in Subfields of Biomedical Research

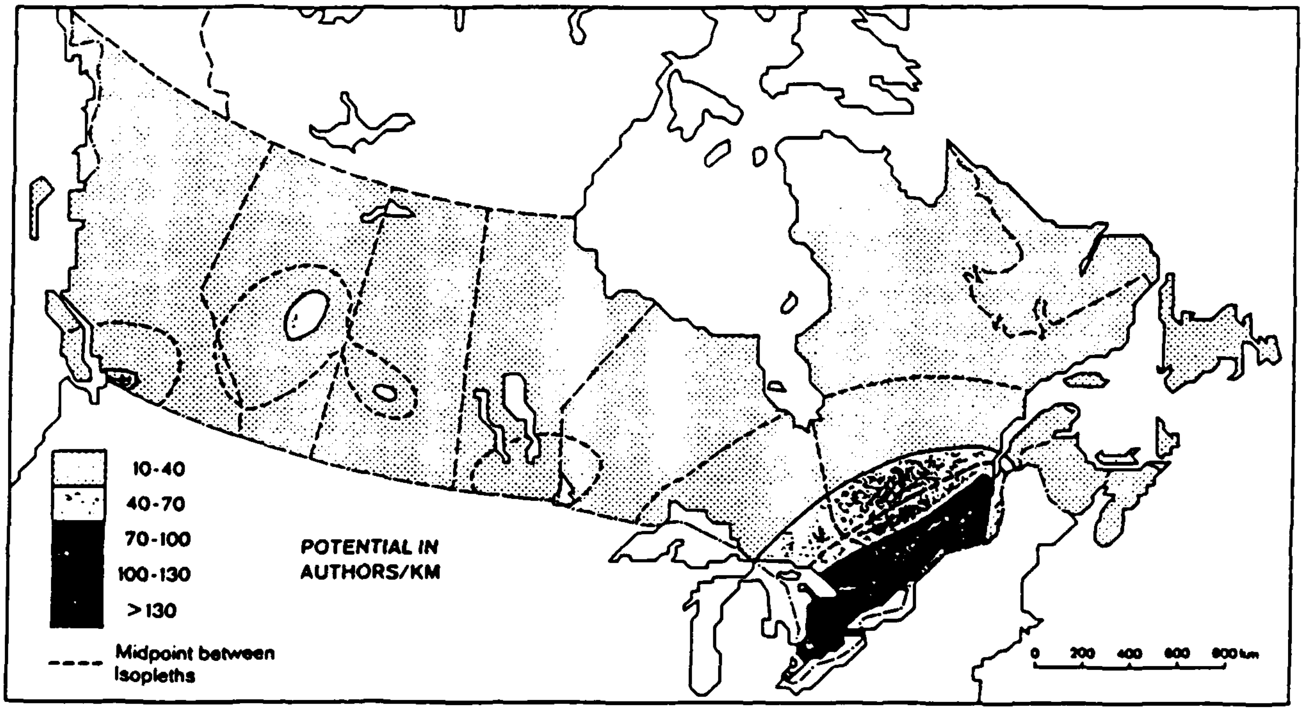
240 BID–MEDLINE Journals
1973–1976 over 1977–1980

$$\text{INDEX} = \frac{\text{\% of Institute's Papers in Subfield}}{\text{\% of 240 Journals' Papers in Subfield}}$$



Note: Scientific Fields or Subfields with less than 1 percent of papers for this BID in the given year range are shown as open bars.

FIGURE 10: DISTRIBUTION OF CANADIAN SCIENCE



(from Inhaber and Przednowek, 1974)

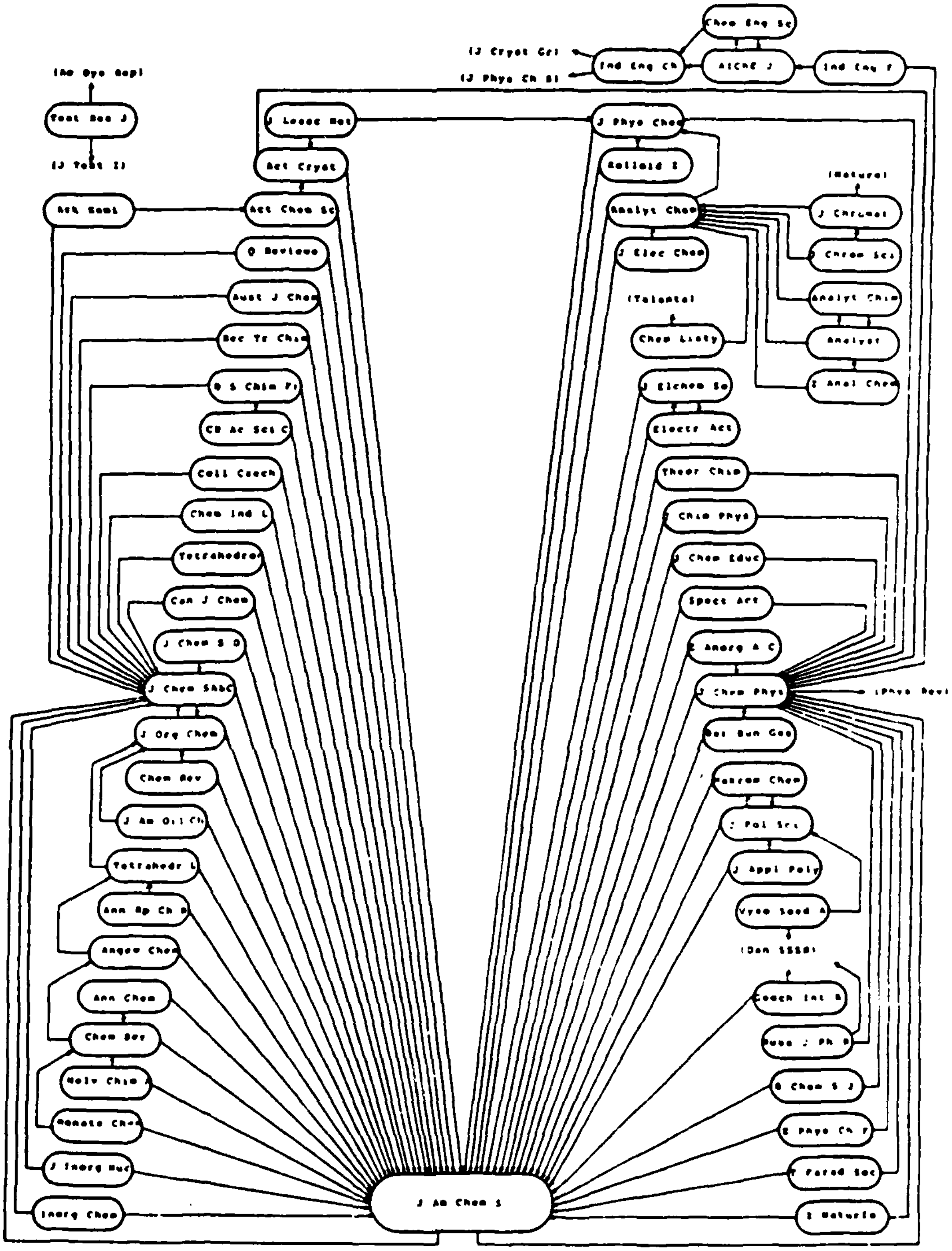


FIGURE 11

TWO-STEP MODEL FOR 62 CHEMISTRY JOURNALS IN 1969
(from Narin, Carpenter & Berlt, 1972)

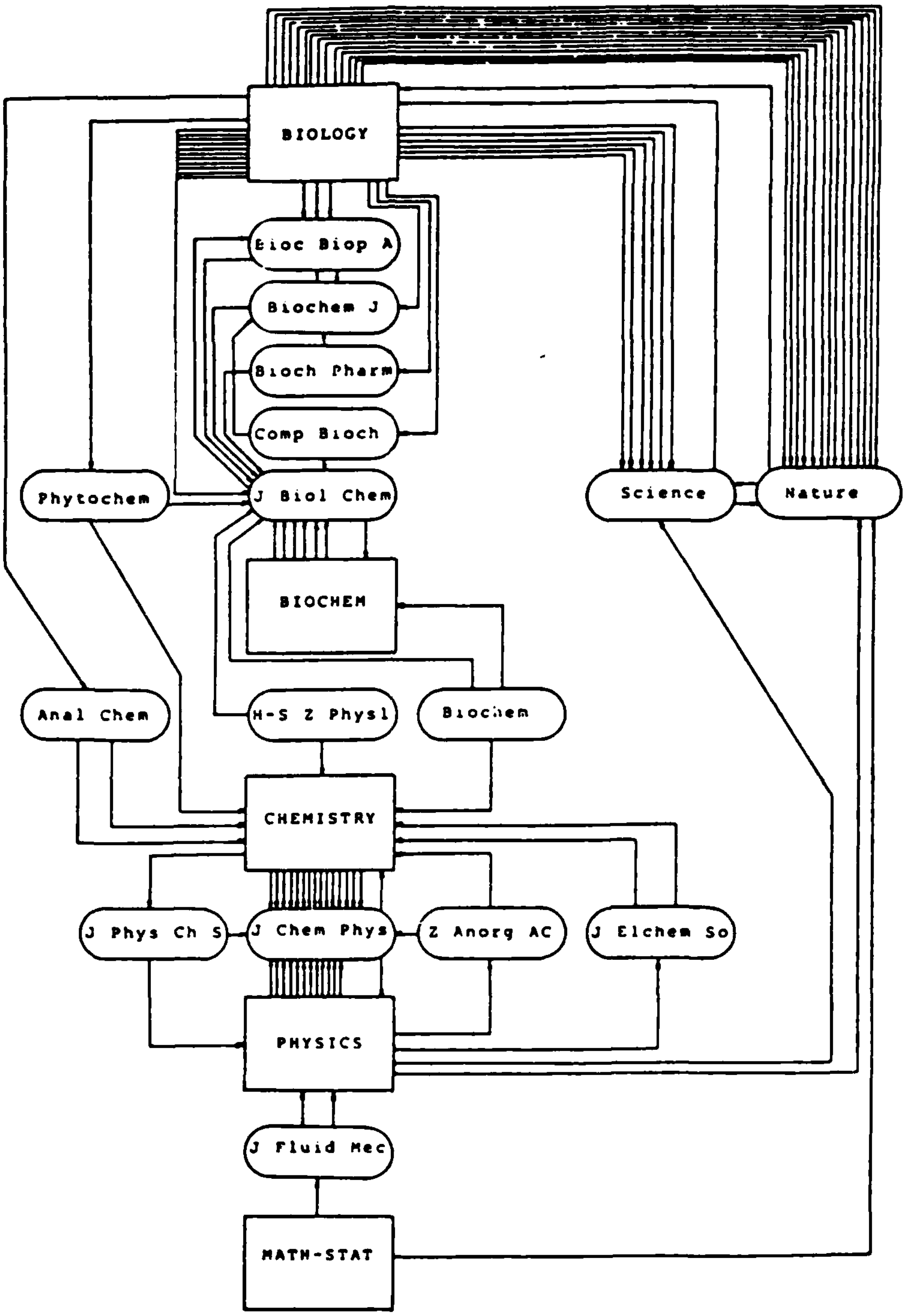
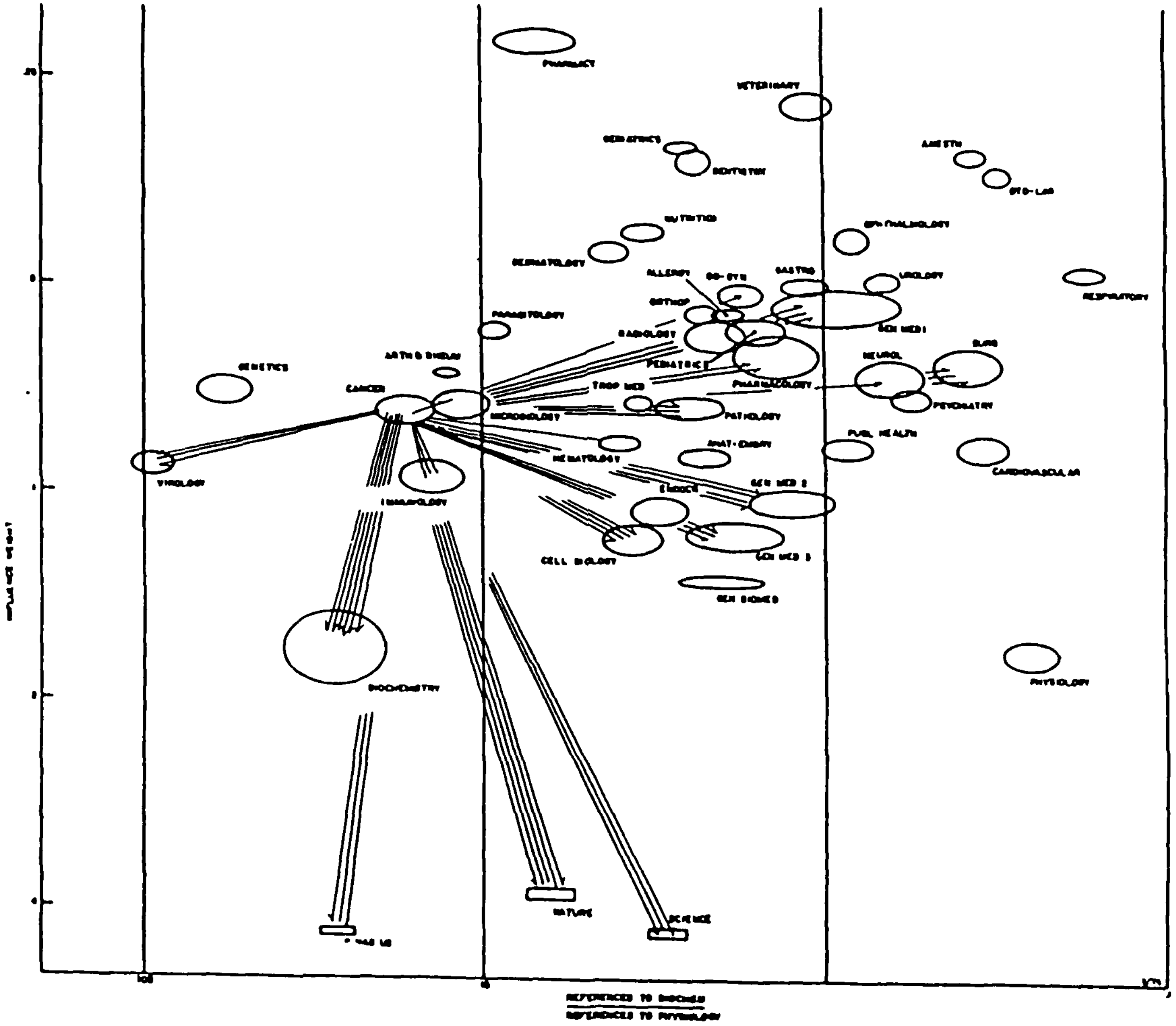


FIGURE 12

TWO-STEP CROSS-FIELD MODEL - 1969
(from Narin, Carpenter & Berlt, 1972)

FIGURE 13: REFERENCING FROM CANCER JOURNALS



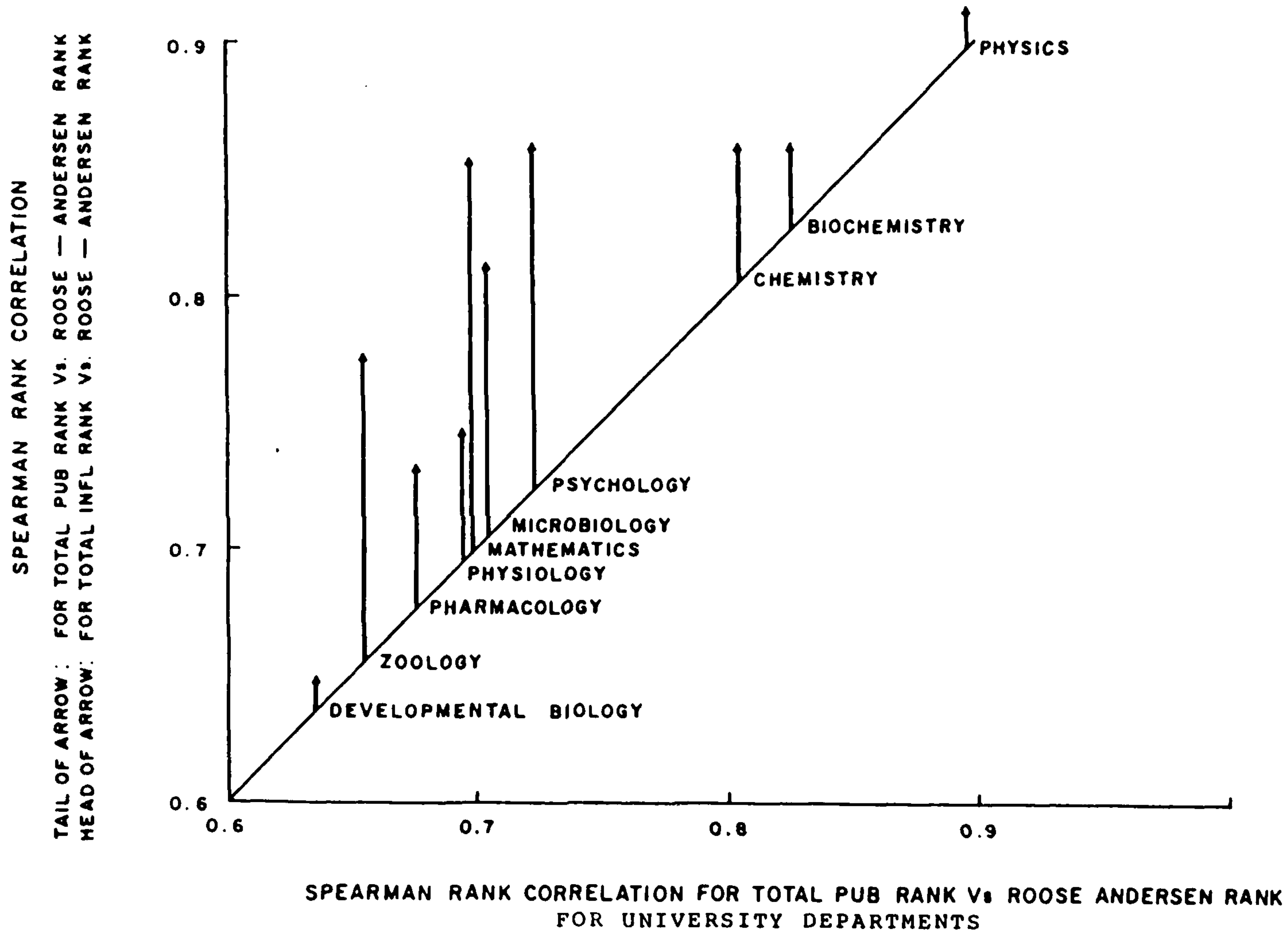
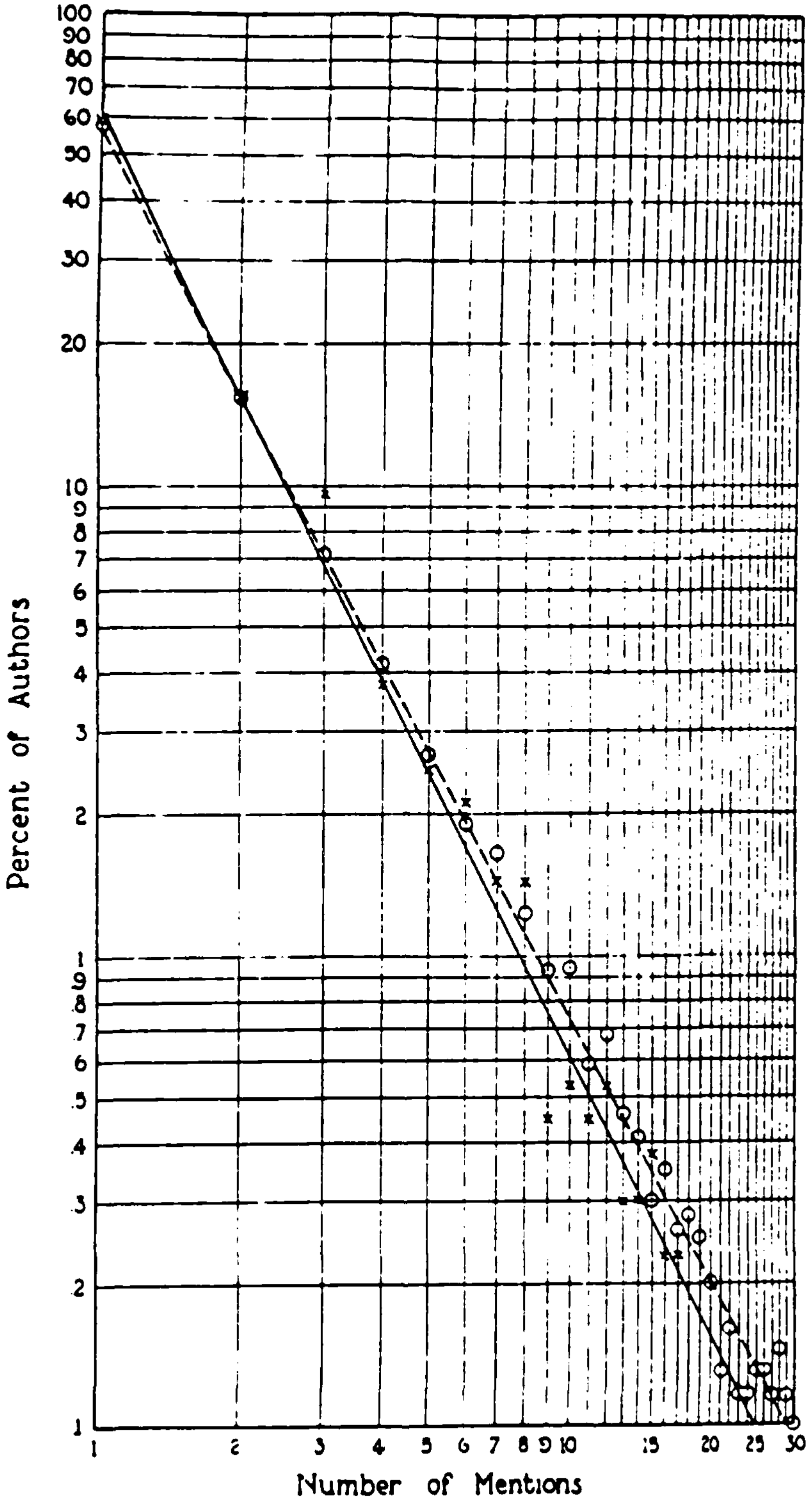


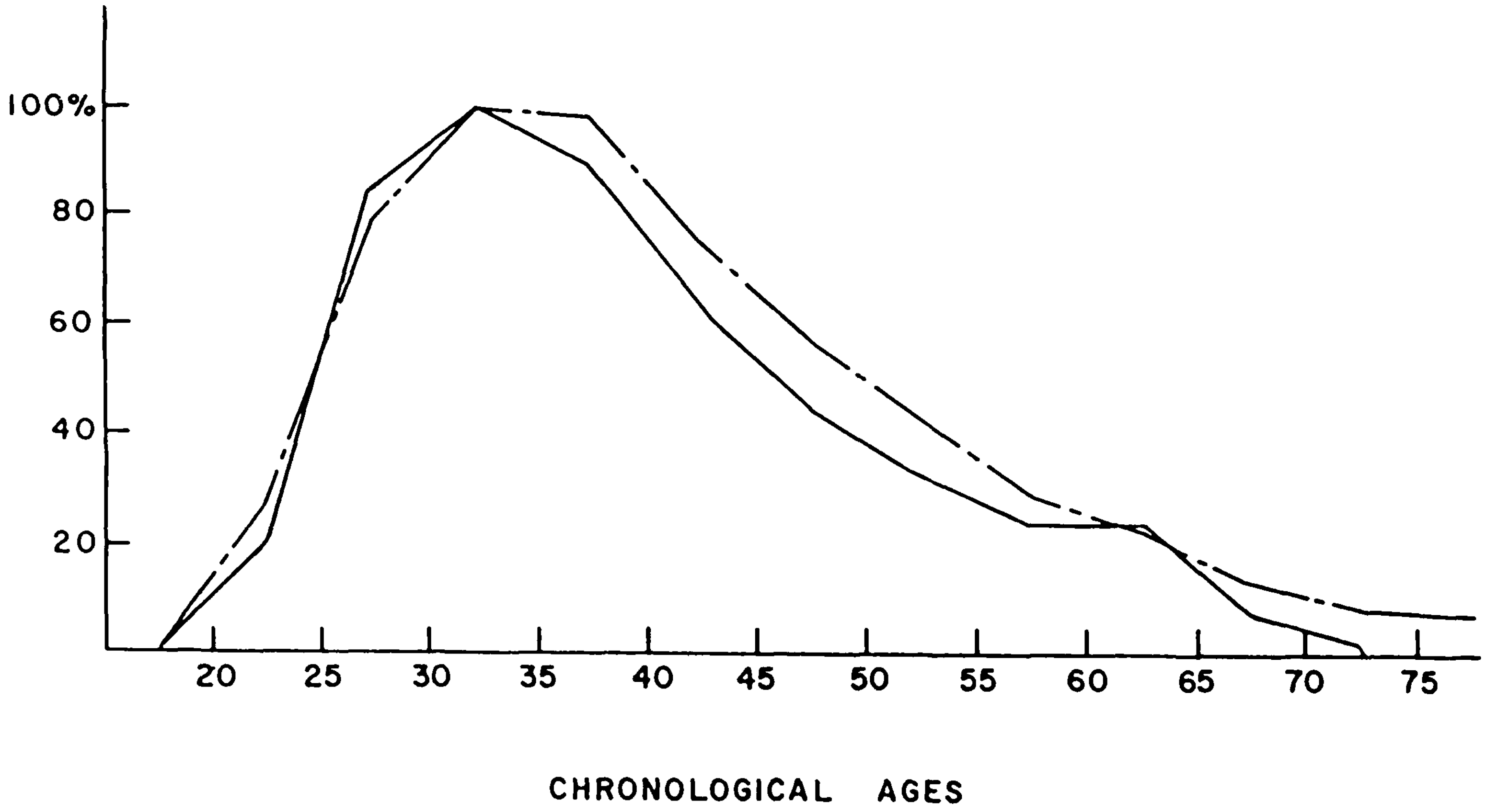
FIGURE 14: SPEARMAN RANK CORRELATION FOR PUBLICATION AND INFLUENCE WEIGHTED PUBLICATION RANK VERSUS ROOSE-ANDERSEN RANK
(from Anderson, Narin and McAllister, 1978)



Logarithmic frequency diagram showing number of authors mentioned once, twice, etc., in Auerbach's tables (points indicated by crosses), and in Chemical Abstracts, letters A and B (points indicated by circles).

FIGURE 15

PUBLICATION FREQUENCY FOR AUTHORS
(from Lotka, 1926)



Age versus contributions to chemistry. Solid line, median values obtained from 11 statistical distributions (percentage values) of age data for "still-living" contributors. Broken line, comparable age data for 11 deceased groups born subsequent to 1774.

FIGURE 16

AGE VERSUS CONTRIBUTIONS TO CHEMISTRY
(from Lehman, 1962)

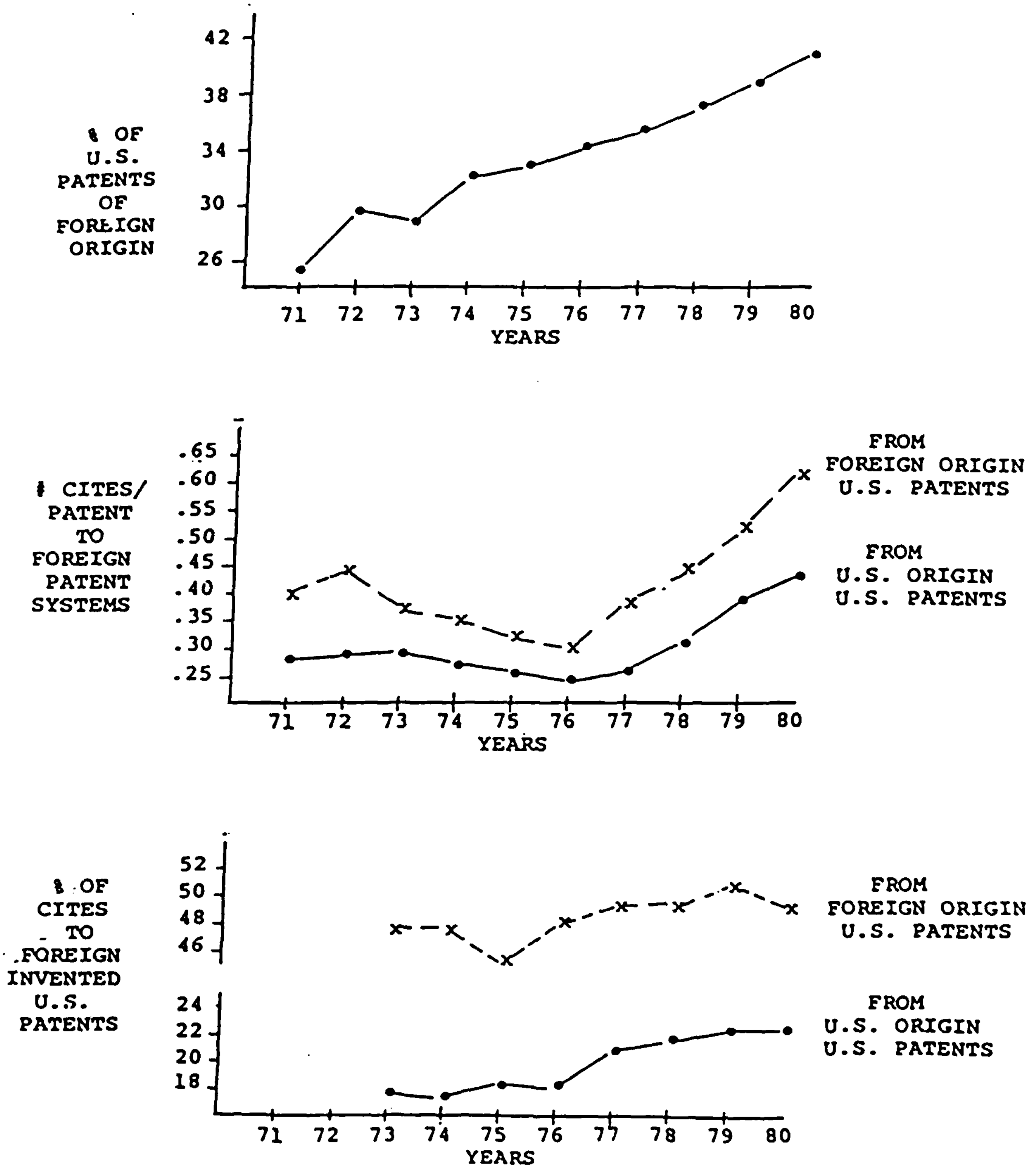


FIGURE 17

FOREIGN DEPENDENCE OF U.S. PATENTS IS INCREASING

IN SCIENTIFIC INSTRUMENTATION

$$\text{INDEX OF SIMILARITY} = \frac{I_{AB}}{\sqrt{I_A}\ \ \sqrt{I_B}}$$

WHERE I_X = INDEX OF CONCENTRATION FOR THE PATENTS OF X

= PROBABILITY THAT ANY TWO PATENTS FROM X
ARE IN THE SAME U.S. PATENT OFFICE CLASS

I_X VARIES FROM 1 (ALL PATENTS IN ONE CLASS)
TO 0 (ALL PATENTS IN DIFFERENT CLASSES)

	MARTIN	BENDIX	ALLIED	HITACHI	IBM	BURROUGHS
MARTIN	1.00					
BENDIX	0.32	1.00				
ALLIED	0.28	0.26	1.00			
HITACHI	0.40	0.39	0.26	1.00		
IBM	0.29	0.26	0.22	0.68	1.00	
BURROUGHS	0.23	0.28	0.25	0.53	0.87	1.00

FIGURE 18

INDEX OF CORPORATE PATENTING SIMILARITY

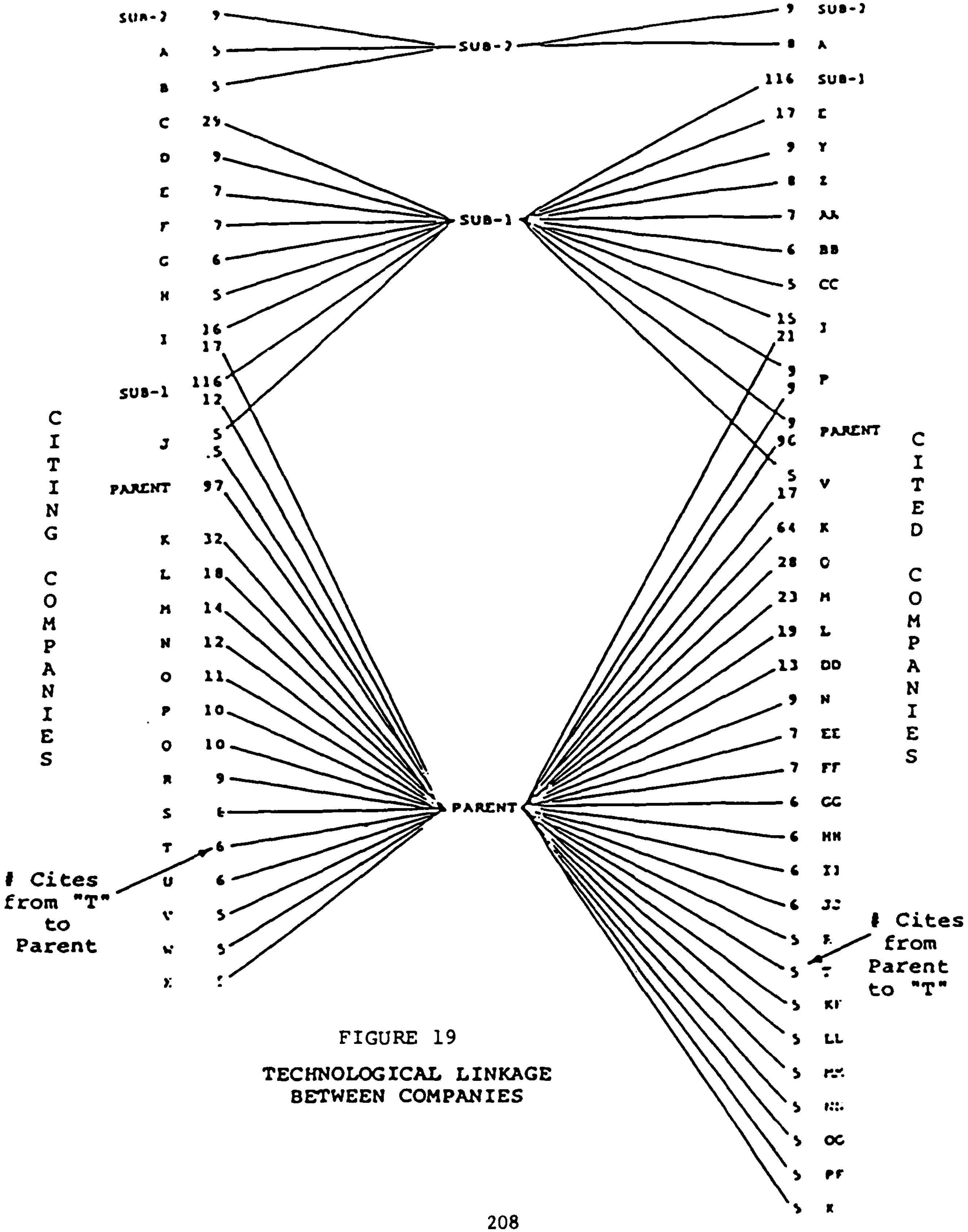


FIGURE 19

TECHNOLOGICAL LINKAGE BETWEEN COMPANIES

FIGURE 20

CITATIONS ON THE FRONT PAGE OF U.S. PATENTS

6 CITATION/PATENT TO U.S. PATENTS

1/3 CITATION/PATENT TO NON-PATENT SOURCES

2/3 CITATION/PATENT TO FOREIGN PATENTS

REFERENCES TO NON-PATENT SOURCES

37% TO SCI JOURNALS

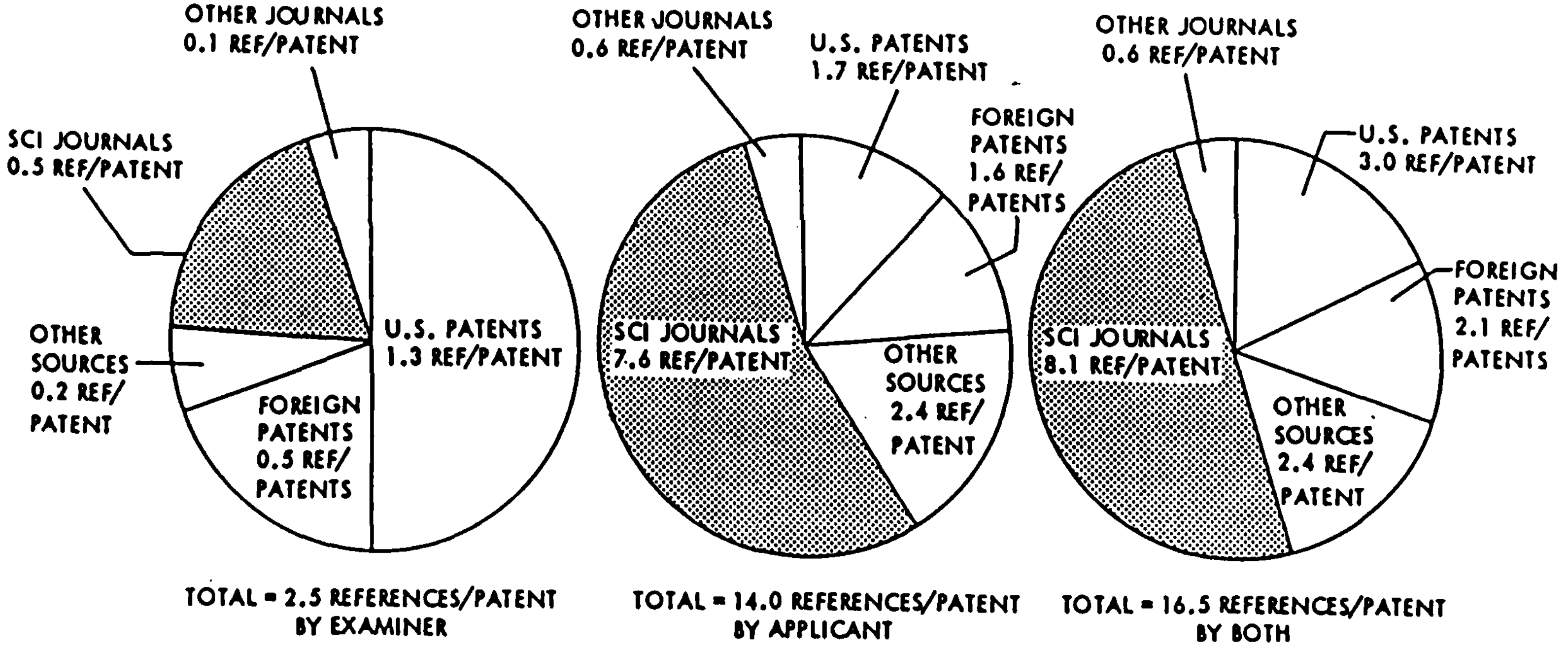
11% TO OTHER JOURNALS

15% TO BOOKS

11% TO ABSTRACTS (EXCEPT CHEM ABSTRACTS
INCLUDED IN OTHER CATEGORIES)

FIGURE 21

DISTRIBUTION OF 6593 REFERENCES CONTAINED IN 399 PROSTAGLANDIN PATENTS



(from Carpenter, Cooper and Narin, 1980)

FIGURE 22

CITATION TIME DISTRIBUTION (EXAMINER CITES)

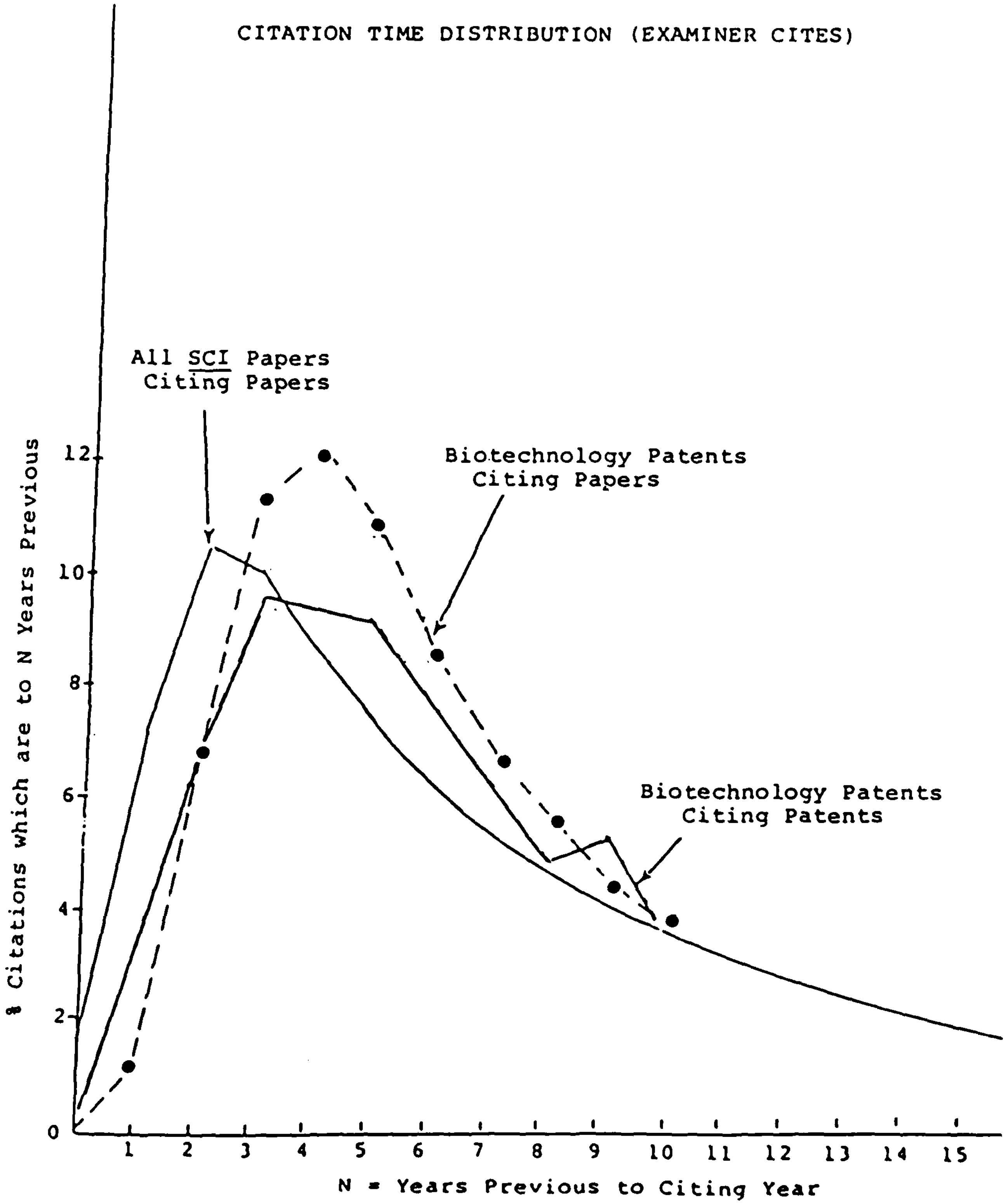


FIGURE 23

CITATION DISTRIBUTION FOR PATENTS AND PAPERS

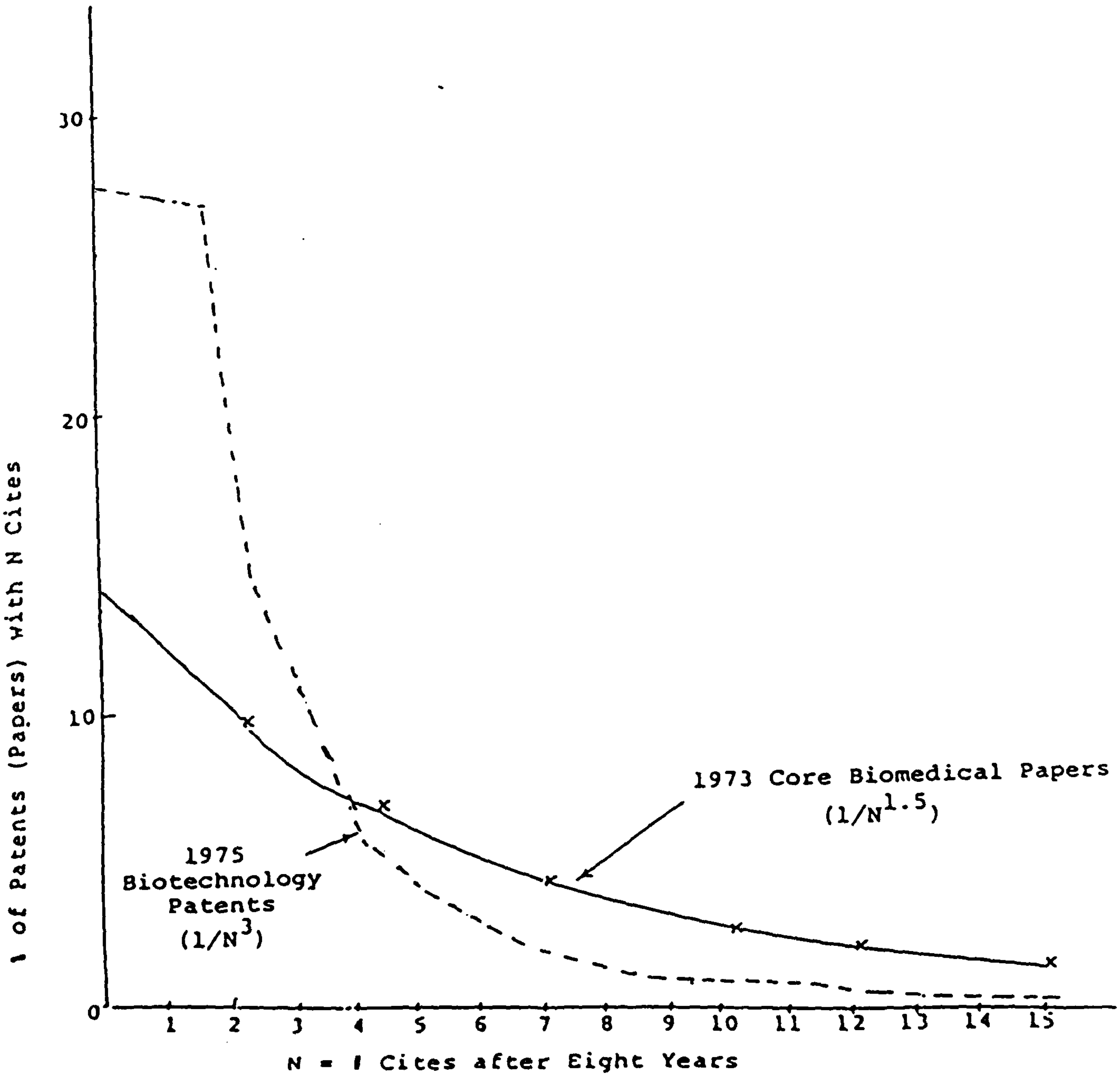


FIGURE 24

TYPE OF PAPER CITED BY SEVEN SUBCLASSES

BASIC RESEARCH (LEVEL 4 JOURNALS)	28%
APPLIED RESEARCH (LEVEL 3 JOURNALS)	39%
ENGINEERING & TECHNOLOGY (LEVEL 1 & 2 JOURNALS)	33%

THIS IS, ESSENTIALLY, EVEN CITING ACROSS THE BASIC-TO-APPLIED RESEARCH SPECTRUM

FIGURE 25

CONCLUSIONS

1. FUNDS AND SCIENTIFIC PUBLICATION ARE HIGHLY LINKED AT NATIONAL AND INSTITUTIONAL LEVELS

2. CROSS-NATIONAL, CROSS-INSTITUTIONAL GEOGRAPHIC AND SUBJECT-TO-SUBJECT LINKAGES ARE QUANTIFIABLE AND HIGHLY CORRELATED

3. CITATION FROM PATENTS TO PAPERS DOCUMENTS A CLOSE, DAY-BY-DAY RELATIONSHIP BETWEEN SCIENCE AND TECHNOLOGY

4. THE SCIENCE USED BY PATENTED TECHNOLOGY IS VERY RECENT--IMPLYING LITTLE OR NO LAG BETWEEN DISCOVERY AND UTILIZATION

5. TECHNOLOGY, LIKE SCIENCE, MAY BE DRIVEN BY A SMALL NUMBER OF KEY, HIGHLY CITED EVENTS

THE "WORLD BRAIN" TODAY: SCIENTOGRAPHIC DATABASES.

WHAT ARE THEY, HOW ARE THEY CREATED AND WHAT ARE THEY USED FOR?

George Vladutz

ISI — Institute for Scientific Information

Philadelphia, Pennsylvania

THE "WORLD BRAIN" TODAY: THE SCIENTOGRAPHIC APPROACH TO KNOWLEDGE REPRESENTATION

George Vladutz

Institute for Scientific Information

Introduction

In his introductory remarks to this conference, Dr. E. Weinberg pointed to some of the problem areas in which one can expect benefits from the modern data base management technologies now under development (e.g., see [1]). The assumption of this paper is that one of the most important of these areas is the global productivity of R&D activities in science and technology as well as the increased efficiency of usage of the acquired portions of scientific knowledge especially in domains outside the subfields where they were originally harvested. The ever deepening level of the scientific endeavor may be a serious threat to its global productivity. One reason for this is the increased degree of specialization resulting in diminished possibilities for cross fertilization between even the close enough specialized subfields of knowledge and certainly much less opportunities for the productive interaction between the more remote domains. In practical terms these problems are compounded for the free societies by the fact that such societies are significantly surpassed by the existing totalitarian ones in respect of the human resources they are ready to devote at present to the pursuit of science.

The point this paper will try to make is that for the above reasons it is important to explore the applications of computerized knowledge management technologies to as broad as possible domains of scientific knowledge, and to attempt to get as close as possible to the global scale. In this, the approach we will be exploring differs in important respects from current trends in knowledge engineering, embodied in the highly specialized expert systems [2-4] currently created in the field of artificial intelligence (AI).

The subject of the paper has much to do with the topic of "knowledge representation" [3-5]. By now this is a widely discussed household item of AI research. At the present stage as well as in the predictable future of the development of current trends in AI, knowledge representation techniques are aimed at rather limited fragments of the universe of knowledge. The very limited width of the domains is compensated by the "logical depth" of their treatment. For the different approach to the knowledge representation problem concerned with very broad fragments and even the whole universe of scientific knowledge we suggest the term "scientography" in order to stress the intention to study the "geography of science," as opposed to its in depth analysis. Most of the current research related to the scientographic approach is conducted in conjunction with efforts aimed at "mapping of science." The alternative term "scientography" we propose is meant to juxtapose such efforts with the domain they originated from, namely with "scientometrics," the quantitataive evaluation of scientific activities [6]. Between AI's current approach to knowledge representation and the scientographic approach there are besides width of scope essential differences concerning their tools and goals.

When constructing specific expert systems a certain amount of necessary common sense knowledge, as well as of detailed domain specific knowledge is extracted from the expert's heads by a joint effort with AI specialists. After due formalization such extracted "live" knowledge, consisting of facts and heuristic reasoning procedures is input into the computer in order to solicit from it problem solving capacities. Such capabilities are then limited to the given narrow domain. In the scientographic approach we try to gain without human intervention a certain amount of generalized knowledge with as little intellectual effort and as much computer assistance as possible. For this we are using exclusively the body of scientific knowledge prerecorded in scientific publications. In this way macroscale knowledge pertaining to important features of the "knowledge landscape" is obtained initially inside of a computer. Afterwards this generalized "macroknowledge" dealing with more or less broad areas of science is exteriorized in forms appropriate for human problem solving. An important type of such human usage of macroknowledge involves "navigation" through the intricate maze of overlapping subfields of knowledge in search of recorded knowledge relevant and useful for solving some given scientific or practical problem. In this respect the usage of scientographic analysis may be a prerequisite for building an expert system geared toward a specific problem domain. Macroknowledge obtained by scientographic analysis can also be used for the measurement, evaluation and management of scientific activities. For many purposes the most convenient form of macroknowledge for human use proves to be a two- or three-dimensional graphic representation, a "map" of the knowledge structure.

What relates together the above two different micro- and macro-oriented approaches to computerized knowledge management is the remote enough, but ultimately common goal: to achieve new problem solving capacities in broad fields of scientific R&D activities. This goal could be potentially achieved either by some gradual broadening of the domain of activities of expert systems (and in this case we will talk about a "bottom-up" approach), or by the gradual "deepening" of the relatively shallow level of macroknowledge obtained in the scientographic approach (in this case we will be dealing with a "top-down" approach to knowledge representation). Nevertheless, we want to argue that it is hard to expect the uncoordinated spontaneous development of expert systems to evolve in the desirable direction unless such development will be guided early enough by the top-down insights provided by the scientographic approach. The convergence of the micro- and macro-approaches to knowledge representation can result ultimately in a broad in scope body of formalized knowledge, structured in a multilevel manner, and equally rich in its "factual" as well as "reasoning" components. Our aim here is to point to some kind of middle-ground between the knowledge representation practices of expert system and present activities in the domain of the structural analysis of science which can facilitate such integration and convergence.

One of the important goals of the scientographic approach, to facilitate the efficient usage of a maximum amount of scientific knowledge, is closely related to the concept of the so-called "World Brain." This title of a volume of essays by H.G. Wells, published in 1938 [7], was later used in several essays and talks by E. Garfield [8-10] in which he discussed the ideas concerning the evolving prospects of global information banks. In the same period when Wells' book was published the concept of an "International Unified Encyclopedia of Science" was proposed by such prominent representatives of logical positivism as R. Carnap and O. Neurath [11]. Our approach to computerized knowledge management has high affinity with this latter vision because of the theoretical (logical) underpinning we envisage (and will briefly describe below) for scientographic techniques. In terms of goals and methods we see also an essential relatedness to M. Kochen's concept of the World Information Synthesis and Encyclopedia (WISE) [12] and its latest derivative, the idea of a system called EUREKA-1 [13], specially aimed at facilitating communication between duly matched scientists in order to create the best links conducive to new discoveries.

The idea of the development of "integrated world knowledge systems" as a result of the further developments of AI research was mentioned recently by R. Schank [14], who thinks this can happen "within the next 50 years." According to Schank, "the most effective role for these systems will be as librarians and consultants."

AI specialists believe that: "As an inevitable side effect, knowledge engineering will catalyze a global effort to collect, codify, exchange, and exploit applicable forms of human knowledge" [2]. This paper will try to call the attention of knowledge engineers to some methods developed outside of the field of AI on which such a global effort can be based. We believe these methods will ultimately become part of the arsenal of tools of knowledge engineering.

1. Basic Approaches to the Problem of Knowledge Representation

The most important common goal of AI researchers working on knowledge representation problems is the formalization of common sense reasoning complemented by the specific basic facts and heuristic reasoning procedures used in some particular narrow domain of expertise. The most controversial issue in the theory of knowledge representation, as well as in the practice of what is usually designated as "knowledge engineering," seems to be the choice between two types of very different looking KR tools, M. Minsky's frames [15] and the language of predicate calculus (PRC). (For a good introduction to the classical form of PRC with examples of its applicability to knowledge representation problems, see the forgotten by many book by R. Carnap [16].) The classical predicate calculus is also proposed in more appropriate for real life conditions non-monotonic variants [17-18] which are being successfully developed at present.

Insofar as the goals of the scientographic approach are concerned, this choice does not seem to be too difficult. At least from a theoretical point of view, it seems that frames and scripts are more or less simply expressible by means of PRC [19] whereas the reverse is certainly not feasible. Below are some relevant quotations of views expressed four years ago [5]: "Ninety percent of what is done in the representation of knowledge is reinvention, most frequently of predicate calculus" [20]. "There is only one language suitable for representing information . . . and that is first order predicate logic" [21]. And finally a third quotation representing the well-balanced point of view of some authoritative metaexperts: "Although the issues for reasoning in expert systems go beyond those of classical logic, a knowledge of predicate calculus in an essential foundation for understanding issues of representation and inference" ([2], p. 61).

There is certainly full consensus among all AI specialists that the possibilities for the modeling by computers of the reasoning functions of the human brain are ultimately based on the results of symbolic logic. One of the most elementary and at the same time most important findings of symbolic logic is that for the mechanical modeling of inference procedures knowledge has to be translated from its natural language form, in which it is communicated between humans, into some specially simplified artificial form of such languages. The most common of such knowledge representation tools is in the language of first order predicate calculus (PRC). Such PRC language is in essence nothing more than a kit of a few different types of standard forms which, like empty molds, can be used to contain the components of any specific body of knowledge. Presumably these forms correspond in some important ways to basic features of the human cognition.

Another type of structure, namely associative networks of concepts, seem to play an important role in the de facto mechanisms of human thinking. Since associative links between distinct elements of knowledge are easily described by graphs, which in turn are structures we have vast experience to handle by computers, it is natural that much attention has been devoted to examine the possibilities of the so called "semantic networks" as tools for knowledge representation [22-23]. At first the nature of the linkages between the concepts placed in the nodes of semantic networks was limited to the taxonomical "genus-species" relationship labelled as the "IS-A" type of connection. Gradually the nature of such connections as well as the nature of the objects placed in the nodes has evolved in many different ways toward complexity [24]. At present it seems important to clarify the foundations of the usage of semantic networks as knowledge representation tools but there is no doubt that such foundations also rely ultimately on PRC. An important step in this direction was made by Sowa [3] who, based on Pierce's earlier suggestions, has developed a graph notation for logic, more natural than standard logic. In the next section we will try to illustrate on an elementary level the role of PRC using standard PRC notation and some simple examples.

As will be seen later, the scientographic approach is aimed at providing a knowledge representation tool which is in essence some kind of semantic network of mid-way strength which can be constructed for large domains of knowledge with clearly defined connections with the significantly stronger formalized representations the application of PCR based methods of the type of conceptual graphs may ultimately provide.

2. An Example of Knowledge Representation

According to [2], "knowledge consists of (a) the symbolic descriptions that characterize the definitional and empirical relationships in a domain and (b) the procedures for manipulating these descriptions." PCR-type tools play important roles in the computer representation of both these knowledge components. Our goal will be first to illustrate on an elementary level the ways a PRC system can be used to achieve these purposes. In the process we will attempt to give an idea about the standard form of the language of PRC to those readers who don't happen to be familiar with its symbolism.

Our examples of knowledge representation will be taken from a specific subfield of information science called "citation analysis," which deals with some facts involving scientific publications, e.g., facts concerning the citation of certain publications by other publications, the co-citedness or the bibliographic coupling of certain pairs of publications and the conclusions one can draw from such facts. This specific choice of knowledge domain is motivated by two different reasons. First, it is expected that most of the readers, as users of the scientific literature, are vaguely familiar with the domain but are not familiar with the meaning of some of the above terms. That circumstance makes it easier to illustrate the role of the formalization of knowledge in connection with the creation of knowledge representation tools of the type we want to illustrate. Secondly, as it will be argued later, the methods of citation analysis happen to be among those which can contribute to the creation of knowledge representation tools relevant to the scientographic approach.

Let's start by describing the basic features of a PRC language in its most traditional form.

The statements describing the facts, i.e., the state of affairs of the world are expressed in a PRC by combining the (arbitrarily chosen) denotations for the **individual objects** we can observe, or can think about, with **predicates**, which are descriptions of different types of relations of "situations" these individual objects can engender. Each

predicate corresponding to a specific type of configuration or relationship has a specific number of empty slots (places) in which the names (symbols) of the participating **individual objects** can be inserted.

In our examples, we will be dealing with objects of the type of **texts**. Specifically we will consider texts of two different sorts: scientific publications (called also documents) and bibliographic descriptions of scientific documents (called also references). We chose to designate these types of objects by the symbols $\mathbf{d}_i$, $\mathbf{d}_j$, $\mathbf{d}_k$, ..., etc., (for documents) and $\mathbf{b}_r$, $\mathbf{b}_s$, $\mathbf{b}_t$, ..., etc. (for bibliographic descriptions). As examples of predicates, we will take the 2-place predicate **BDS(,)** meaning "The given text (first slot) is the bibliographic description of certain specific publication (second slot)" and the 2-place predicate **PTXT(,)** meaning "The given text (first slot) is part of a (longer) second text (second slot)." Using these symbols, we can now give examples *of prepositions* like: $\mathbf{BD}(\mathbf{b}_g,\mathbf{d}_i)$, equivalent to the natural language statement, "the bibliographic description $\mathbf{b}_g$ is part of the text (of the) document $\mathbf{d}_j$." Such logical propositions can be false or true. Following the precepts of the situational semantics as proposed by J. Barwise and J. Perry [25], we will consider as the interpretations of such sentences the real world <u>situations</u> described by them.

Using the above denotations for the concepts of "bibliographic description" and "part-whole" relationship for texts, we will introduce **definitions** for some other concepts from the domain of citation analysis. In all the following formulas, the variables **x, y, z**, ..., etc., are variables for documents and **t, s, v**, ..., are variables for bibliographic descriptions.

$$\mathrm{CIT}(x,y) =_{Df} (\exists s)\ [(\mathrm{BDS})(s,y) \text{ and } \mathrm{PTXT}(s,x)] \qquad (1).$$

This newly introduced predicate (and concept) "publication x cites publication y" means that it exists (symbol " ∃ ") such a reference s that s is the bibliographic description of document y, and that s is part of the (text of) document x".

$$\mathrm{CTD}(x,y) =_{Df} \mathrm{CIT}(y,x) \qquad (2).$$

A publication y is cited by another publication x if and only if publication x is citing publication y.

$$\mathrm{COCIT}(x,y) =_{Df} (\exists z)[\mathrm{CIT}(z,x) \text{ and } \mathrm{CIT}(z,y)] \qquad (3).$$

Two publications are co-cited if there is some (third) publication z such that z cites both x and y.

$$\mathrm{BIBCP}(x,y) =_{Df} (\exists z)[\mathrm{CTD}(z,x) \text{ and } \mathrm{CTD}(z,y)] \qquad (4).$$

Two publications are bibliographically coupled if there is some (third) publication z such that z is cited by both x and y.

The above notations allow us to describe as axioms some domain specific facts we may believe to be self evident. For example:

$$(\forall x)[(\exists t)\ \mathrm{BDS}(t,x)] \qquad (5),$$

i.e., for all (symbol $\forall$) documents (any document x) there is some bibliographic description;

$$(\forall x)[\sim CIT(x,x)] \equiv \sim(\exists x)[CIT(x,x)] \qquad (6),$$

i.e., for all documents (any document x) it is not true (symbol ~) that x cites itself, or equivalently, it is not true that there is such publication x which cites itself;

$$(\forall x)(\forall y)[CIT(x,y) \rightarrow \sim CIT(y,x)] \qquad (7),$$

i.e., if a document x cites document y, then document y is not citing document x. (The symbol "→" is used for the logical implication.)

A PRC system makes it possible to perform inferences using certain universal rules. As an example of such rules, we will consider the so-called "universal specialization rule" which allows to conclude e.g., from "All men are mortal" and "Socrates is a man" that "Socrates is mortal." The rule's symbolic representation (the symbol "$\vdash$" is used for "entails" or "yields"):

$$(\forall x)P(x),\ a \ \vdash P(a) \qquad (8),$$

where a stands for the denotation of any object of the same sort for which the variable x is used for and $P(x)$ is a proposition containing the only free (not bound by any $\forall$ or $\exists$ symbol) variable x.

In order to realize how this inference rule can be used with the above axiom (7), one has to detect that (7) has the $(\forall x)P(x)$ form required in the antecedent of the inference rule (8). Let b_5 be a specific document, to play the role of a in the second antecedent of (8); the application of the rule, which involves the substitution of x by b_5 yields then as consequence:

$$(\forall y)[CIT(b_5, y) \rightarrow \sim CIT(y, b_5)] \qquad (9).$$

This is a proposition of the form $(\forall x)P(x)$, in which y plays the role of x. Therefore (9) taken together with the symbol of another specific document, e.g., b_1, will yield:

$$CIT(b_5,b_1) \rightarrow \sim CIT(b_1,b_5) \qquad (10).$$

This perfectly trivial two step inference led from the universal statement (7) to the fully specific (10).

Suppose that among the facts stored in the knowledge base of an expert system the record

$$CIT(b_1,b_5) \qquad (11).$$

can be found and suppose the question "Is document b_1 cited by document b_5?" is asked from the system. First, this query is translated by a natural language interface into the proposition

$$CTD(b_1,b_5) \qquad (12)$$

to which the status of hypothesis is assigned. Secondly, the system initiates inference procedures of the kind of the above example in order to generate propositions related to the content of the query. A first step of such procedures can be the simplification of expression by the substitution in it of some derived (non-elementary) predicates by some simpler ones. Using the above definition (2), our query is transformed to the equivalent form

$$\mathrm{CIT}(b_5,b_1) \qquad (13).$$

The specific inference chain $(7)\vdash(9)$, $(9)\vdash(10)$ we described above which yields (10) now can be continued using another universal inference rule of the propositional calculus called "modus ponens." This rule is formulated as: $A \rightarrow B, A \vdash B$, where A, B are propositions. Taking as antecedents proposition (10) and our transformed query (13), we conclude according to this rule: $\sim \mathrm{CIT}(b_1,b_5)$. This proves to be a direct negation of one of the facts stored by the system as record (11). Therefore the query hypothesis is retracted and the answer "NO" to the query is given by the system. It may be the case that axiom (7) is not believed as always true; instead the experts could assign to it a certain high (e.g., 99%) degree of probability with which it applies. In such a case the usage in an expert system of this belief statement will induce a degree of probability to the consequences inferred from it. Therefore the system, instead of answering "NO" may produce the answer "Very likely, no."

This very simple and purely illustrative example can give only a superficial idea about the role which, in principle, PRC-type deductive tools play in expert systems.

The specific ways in which inference rules can be efficiently put to use are those which happen to be essential from a practical point of view. Important for this purpose are some best strategies (heuristic rules) determining the choice of universal inference rules and the order of their application to the appropriate antecedents in order to avoid the combinatorial explosion of non-relevant useless consequences. Most efficient can be domain specific heuristic rules which can provide efficient shortcuts to the exhaustive searches for solutions. At the present stage of development of expert systems, such domain specific "rules of thumb" often are given the form:

"IF < antecedents > THEN < rule strength or weight > < consequent >."

Such rules can be not only of procedural nature, i.e., conveying efficient inference patterns but can also express factual knowledge about causal or other important empirical dependencies between situation types of the given domain. In both cases one deals mainly with the domain specialist's private knowledge "that has not found its way into the published literature" [2].

3. Concept Dependency Diagrams as Knowledge Representation Tools

Inferential efficiency can certainly make the difference between the practical feasibility and the impossibility to solve a given problem with a computer. Another not less important type of limitation one has to face when dealing with an expert system is its limited domain of expertise as determined by the scope of its knowledge base. A good general way of describing the scope of a problem solving system is to list the concepts used in the description of facts, beliefs and procedural rules constituting its knowledge base. If a problem is formulated in terms of concepts which a system cannot handle, the problem remains outside the system's domain of expertise, unless the user can reformulate the problem so that those concepts can be expressed in terms of the system's concepts. The set of concepts available to an expert system is completely defined when the system is built and this set usually remains its most stable feature even when the system's knowledge base is expanded or updated. The facts and beliefs recorded in the knowledge base create the specific set of links between concepts which determine the set of potentially inferable consequences. Such consequences have to express certain other specific concept links required by the conditions of a problem in order to be its solution. If there are no adequate means in the knowledge base for creating the linkages between the concepts specified by a problem, then this problem cannot be solved in the system. Therefore, the degree of intelligence of an expert system can be described in a general way by listing concept pairs for which linkages are stored or inferred inside the system. Using the elementary examples of the previous section, we will examine here the picture of concept linkages created by those examples and general ways for representing concept linkages. For our purposes, we will consider concepts as equivalent to predicates in a PRC based knowledge representation tool.

The definitions (2-4) of the previous section create clear dependency linkages between the defined and defining predicates. One way of representing such links is by a directed graph with two kinds of nodes:

concepts (represented in Fig. 1 by boxes) and definitions (circles). The direction of the arrows is from defining to defined predicates.

This hierarchical diagram of Fig. 1 reflects the dependencies created by the specific set of definitions (2-4) we have used. In order to assert that such a diagram indeed represents a correct picture of the dependency relationships between the given set of concepts, we have to examine whether our graph describes all the possible definition linkages which can be established within this set. It can be easily seen that each subordinate predicate can be defined not only in terms of its immediate superior(s) but also in terms of any more distant superior(s) in the hierarchy. For example, **CTD,** instead of being defined through **CIT** can be defined directly using **PTXT** and **BDS:**

$$\mathbf{CTD(x,y)} =_{\mathrm{Df}} (\exists z)[\mathrm{BDS}(z,x) \text{ and } \mathrm{PTXT}(z,x)] \qquad (2').$$

Therefore it remains to be examined whether definitional links can be established for such pairs as:

CTD → CIT, CTD → COCIT, BIBCP → CTD, COCIT → CIT,
BIBCP → COCIT, COCIT → BIBCP,

which are the only remaining pairs for which no superior-subordinate relationships were established, at least in the given directions.

It turns out that **CIT** is easily defined in terms of **CTD:**

$$\mathbf{CIT(x,y)} =_{\mathrm{Df}} \mathbf{CTD(y,x)} \qquad (1').$$

Since by virtue of (1') **CIT** becomes now a superior of **COCIT** one can conclude that a definition creating the link **CTD → COCIT** can be also formulated. Indeed:

$$\mathrm{COCIT}(x,y) =_{\mathrm{Df}} (\exists\, z)[\mathrm{CTD}(x,z) \text{ and } \mathrm{CTD}(y,z)] \qquad (3').$$

All the other definitions between the above pairs prove not to be feasible. Therefore the complete picture of definability dependencies between the predicates of our set is that given in Fig. 2. Here the direct definability link CIT → BIBCP, corresponding to the definition:

$$\mathbf{BIBCP(x,y)} =_{\mathbf{Df}} (\sim\exists\, z)[\mathbf{CIT(x,z)} \textbf{ and } \mathbf{CIT(y,z)}] \qquad (4')$$

implied in Fig. 1 by the CIT → COCIT superior-subordinate relationship is also shown to emphasize its symmetry with the **CTP → COCIT** definition link.

From this figure one can see clearly that predicates CIT and **CTD,** each of which is definable through the other are fully equivalent in respect of their definability relationships with all the other predicates of the set. Such will be the case, in particular, for all predicates which differ only by the order of their argument slots, but describe the same type of situation. In similar cases of two or several reciprocally definable predicates, it is reasonable to consider them as expressing the same concept. Therefore a group of such predicates can be represented by one node of the diagram. At the same time we can simplify the diagram of dependencies by eliminating the special nodes for the definitions. The resulting definability diagram is shown in Fig. 3.

One of the most important steps in creating a PRC based knowledge representation tool is the choice of the elementary, for the given system, undefined concepts to be used for defining all the other concepts the system will deal with. The choice of specific symbols for these and all the other predicates is only a matter of convenience, either from the point of view of computer processing or human mnemonics.

The list of undefined predicates in the illustrative tiny fragment of an applied PRC language we sketched for the domain of citation analysis includes only two predicates: **PTXT(x,y)** for the part-whole relationship for texts and **BDS(x,y)** for the concept of bibliographic description of a scientific publication. On the diagram of Fig. 3, the **PTXT** and **BDS** nodes are the only ones which don't receive "incoming" arrows, corresponding to definability. In reality, talking about the domain we also used in an implicit way the undefined concepts of "text" and "scientific publication" by choosing different kinds of variables for the different sorts of objects which were acceptable for forming propositions with these predicates. In order to make explicit the usage of these concepts, we will add two more one-place undefined predicates to the list: **TXT(x)** for the concept "IS-A-TEXT" (the object whose symbol is filling the slot of this predicate has the property of being a text) and **SPBL(x)** for the concept "IS-A-SCIENTIFIC-PUBLICATION" ("**x** has the property of being a scientific publication"). When needed we can now express that a certain variable stands for an object of some specific type using propositions or propositional forms based on these predicates instead of imposing limitations on the symbols used for variables.

The predicates **TXT** and **SPDL** will remain undefinable, i.e., elementary for our system, as are **PTXT** and **BDS.** In order to represent in the concept dependency diagram all the concepts we are dealing with in our domain, we have to express the important semantic relationships we are aware of as existing between our elementary predicates in virtue of the meaning which we have chosen for them. Such self-evident semantic relationships between undefined predicates can be described by axiom-type propositions we call "meaning postulates":

$$(\forall x)[\mathbf{SPBL}(x) \rightarrow \mathbf{TXT}(x)] \qquad (14),$$

$$(\forall X)(\forall y)[\mathbf{PTXT}(x,y) \rightarrow (\mathbf{TXT}(x) \ \text{ and } \ \mathbf{TXT}(y))] \qquad (15),$$

$$(\forall x)(\forall y)[\mathbf{BDS}(x,y) \rightarrow (\mathbf{TXT}(x) \ \text{ and } \ \mathbf{SPBL}(y))] \qquad (16).$$

Meaning postulates (14) and (15) express the fact that the concept of scientific publication, as well as of the part-whole relationship between texts, presupposes the concept of text and allows us to infer the property of being a text from the fact that an object is a scientific publication or from its participation in a "part-whole" type of situation for texts. Similarly postulate (16) shows that the concept of bibliographic description presupposes both concepts of text and scientific publication and allows us to infer the corresponding properties for any participants of a **BDS**-type of situation. In virtue of the definitions given to concepts **CIT**, **CTD**, **COCIT**, and **BIBCP** the property of being a publication, and therefore also a text, is inherited for all the participant objects of the corresponding situations.

The dependencies between our elementary concepts expressed by meaning postulates are similar in nature to the definability dependencies and can be represented in the same diagram. The resulting picture of dependencies, including those established by definability relationships as well as those expressed by meaning postulates is represented in Fig. 4.

We will introduce some more undefined predicates in order to be able to describe important facts from the domain of citation analysis which make it relevant to the scientographic approach to knowledge representation. The presupposition we will assume is that some texts are meaningful, i.e., have a content which can be translated at least partially into some PRC-type language. The property of being a predicate of such a language, used to record in a formalized way the content of texts, we will designate in our system by the one-place predicate **CPT( )**, meaning "IS-A-CONCEPT." We have to take note of a special circumstance when introducing this symbol: unlike it was the case for the rest of the predicates we used until now, the object whose symbol can be used to fill its slot in order to form a proposition of our formalized PRC-language is not one of the objects of the world existing independently of our formalized language, such as texts of publications, bibliographic descriptions of publications, etc., but is in itself a symbol of our formalized language. For instance, if we want to state the fact that the predicate BDS we used above as a symbol of our

PRC-language is indeed a predicate symbol, and therefore — in accordance with our understanding — it is the equivalent of a concept, we would need another symbol which could enable us to refer to this symbol in its quality of a character string of a text written in our formalized language. The regular method by which such names are formed is the inclusion of the original character string (e.g., BDS) in quotation marks (**"BDS"**). Thus, the five character string "BDS" becomes the name of the three character string BDS, occurring in some text written in our formalized language. Such expressions are meant to be used in the formalized metalanguage, i.e., the portion of our formalized language we will use to formally record facts about its own expressions and about the relationship between such expressions and the situation of the outside world. Then the expression **CPT("BDS")** will mean that -- according to our understanding -- the string BDS is a predicate and therefore the equivalent of a concept: in this particular case we are dealing with the concept corresponding to the property of a text to be a bibliographic description of a scientific publication. We can even write CPT("CPT") for stating that CPT is a concept, in this case the concept of a concept. In order to differentiate the predicates used to form propositions about facts of the outside world from the predicates engendering propositions involving the expressions of our formalized language, we will designate the latters as **"metapredicates."**

For expressing the relationship between a natural language text and a predicate (concept) occurring in its translation into our formalized language, we will use the 2-place metapredicate **CPTXT(,)** with the argument of its first slot being a specific predicate and the argument of its second slot being the symbol of an individual natural language text. In our formalized language the propositional form **CPTXT(x,y)** is then the translation of the natural language statement schema "concept **x** is used in text **y**." Evidently we have the meaning postulate:

$$(\forall x)(\forall y)[\mathrm{CPTXT}(x,y) \rightarrow (\mathrm{CPT}(x) \text{ and } \mathrm{TXT}(y))] \qquad (17)$$

We will also introduce the undefined one-place metapredicate **SCPT( )** for the property of being a particular kind of a predicate, occurring in formalized translations of scientific texts: the **SCPT** is for us the formal equivalent of the concept of "scientific concept." Then obviously:

$$(\forall x)[\mathrm{SCPT}(x) \rightarrow \mathrm{CPT}(x)] \qquad (18)$$

$$(\forall x)[\mathrm{SPBL}(x) \rightarrow (\exists y)[\mathrm{CPTXT}(y,x) \text{ and } \mathrm{SCPT}(y)]] \qquad (19).$$

These meaning postulates express that every scientific concept is a concept (18) and every scientific publication makes use of some scientific concept. It is also implied by the meaning of the concepts **SCPT** and **SPBL** that

$$(\forall x)]\mathrm{SCPT}(x) \rightarrow (\exists y)[\mathrm{CPTXT}(x,y) \text{ and } \mathrm{SPBL}(y)] \qquad (20),$$

i.e., for every scientific concept there is a scientific publication in which it occurs.

Another way of explicating the meaning of **CPTXT** is to assume the existence of a finite inventory of concepts, of which the scientific concepts form a separate section, in the form of a list of groups of synonymic words, phrases or other more complicated types of textual configurations such that the occurrence of any of them in a natural language text means the usage of the corresponding concept in the given text. Such understanding of CPTXT does not change the validity of the above meaning postulates.

We will define a 2-place predicate meaning that two scientific publications make use of (i.e., share) some common scientific concept. We will interpret this predicate as describing the situation of subject relatedness of two publications and therefore symbolize it by the acronym SRL:

$$\mathrm{SRL}(x,y) =_{\mathrm{Df}} (\exists z)[\mathrm{SCPT}(z) \ \text{and} \ \mathrm{CPTXT}(z,x) \ \text{and} \ \mathrm{CPTXT}(z,y)] \qquad (21)$$

One should notice that in virtue of this definition we have also:

$$\mathrm{SRL}(y,x) \equiv \mathrm{SRL}(y,x) \qquad (21')$$

but it is not always true that:

$$[\mathrm{SRL}(x,y) \ \text{and} \ \mathrm{SRL}(y,z)] \rightarrow \mathrm{SRL}(x,z) \ ,$$

i.e., the subject relatedness relationship is not a transitive one. We can now formally express one of the most important assumptions (and findings) of the domain of citation analysis in the form of an "IF...THEN" heuristic rule:

$$\text{IF } \mathrm{CIT}(x,y) \text{ THEN } \langle 95\% \rangle \ \mathrm{SRL}(x,y) \qquad (22).$$

In other words, cited and citing documents are, very likely, subject related. (The 95% value for rule strength is used here only for illustrative purposes.) This assumption, put in an informal way, is usually considered as self evident, e.g.,: "Citations are . . . explicit linkages between papers that have particular points in common" [26]. The assumption served as the basis for the invention by E. Garfield of the Science Citation Index [27] as a tool for information retrieval. As he stated: "To understand what is being retrieved in a Science Citation Index search we have to recognize the underlaying concept which is merely symbolized by a bibliographic citation." [28]. This assumption has been confirmed not only by the successful usage of the Citation Indexes for subject searches but also by direct studies (e.g., [29]), which confer to (22) the status of an empirical fact.

Facts create between the concepts participating in their description a type of dependency different in nature from the dependencies created by definability or meaning postulates. This difference is similar to the differences between the analytical (logical) and synthetic (factual) truth. Quine has shown that no sharp distinction could be established between them [30], nevertheless the difference is felt at any given specific stage of development of knowledge in general, or of an individual's knowledge in particular. For example, suppose that when listening to an introductory course of bibliography somebody has just heard that "citation is an instance of the occurrence in the text of a scientific publication of a reference to another publication," and so has acquired the concept of citation as given by definition (1). For this person the statement "Such citations are used to indicate some kind of subject relatedness between the citing and cited publications" (reflecting (22)) will be a new piece of factual information even if heard immedately after the above defining description. Later, however, in the concept schema this person will tend to use the citation concept defined as above and the knowledge about its usual role (purpose) can be so intimately associated that the second above statement (and formula (22) correspondingly) will be felt by him as a meaning postulate or even a part of the modified definition of the concept of citation formulated as: "Citation is an instance of the inclusion in the text of a scientific publication of a bibliographic reference to another publication intended to indicate the subject relatedness of the cited publication to the citing publication," i.e.:

$$\mathrm{CIT}(x,y) =_{\mathrm{DF}} (\quad z)[\mathrm{BDS}(z,y) \text{ and } \mathrm{PTXT}(z,x) \text{ and } \mathrm{SRL}(y,x)] \qquad (1'').$$

A more precise way of describing what is really going on here is to recognize that the factual dependency between the given concepts lead to the formulation of a new concept, in this case the concept of "subject relatedness indicated by a bibliographic reference" corresponding to a new predicate (SRCIT) defined using the predicates connected by the new factual relationship:

$$\mathrm{SRCIT}(x,y) =_{\mathrm{Df}} \mathrm{CIT}(x,y) \text{ and } \mathrm{SRL}(x,y) \qquad (23).$$

In many cases between the defining concepts of a defined concept there is an empirical dependency which is realized prior to the definition.

All the above illustrates that the differences between the three kinds of concept dependencies, as created by definitions, meaning postulates and empirical (factual) relationships are not very sharp.

Therefore, in the overall pictures given on Fig. 5 for all dependencies we discussed the links are represented by equal non-directed lines.

The picture of the dependency relationships between concepts is very much independent from the choice of elementary concepts. In particular, one can easily see that the way we have chosen the elementary concepts in our illustratiave fragment of a PRC language for the domain of citation indexing is neither the only way nor the best way. One could, and strictly speaking one should, go in much more detail when choosing the elementary concepts: elementary predicates can be introduced for the concepts of "author," "author name," journal title," "publication title," "nominal (or de facto) time of publication," etc. In such a case some of our elementary concepts (e.g., the "bibliographic description" concept) would become definable. However, the dependencies between the specific concepts of our system would remain unchanged. Such invariability of concept dependencies is an argument in favor of a significant degree of "objectivity" of the diagrams of the type exemplified on Fig. 5. Such diagrams are suggested as valid tools for the representation of some essential features of the knowledge structure of a given domain.

Obviously, most areas of science share concepts with other areas of knowledge, in particular with those they use as the "prerequisites" as well as with those which use the given field as their "prerequisite." Therefore, in principle, concept dependency diagrams cannot be limited to narrow domains. Such diagrams will have to reflect the sharing of concepts between the areas of knowledge where the concepts are generated and many other areas where they are used. Concept dependency diagrams are therefore knowledge representation tools of a global nature which can depict not only the structure of the conceptual backbone of the different specific domains, but also the intricate interactions between the different fields of knowledge. In respect of their "depth," concept dependency diagrams are less specific than the PRC based knowledge representation tools used in AI systems. Nevertheless, one can perceive a clear correspondence between the concept structures created by the latter detailed knowledge representation tools and the more sketchy structures we advocate as the basis of the global scientographic approach. One should take notice that we use the term "concept dependency diagram" in a sense different from Schank's term "concept dependency graph" [31]. The latter is used for a tool intended for the detailed representation of knowledge.

4. Ideal Ways for the Construction of Concept Maps of the Concept Dependency Diagram Type

Before examining the purely ideal and then the more or less practical ways for constructing knowledlge representation tools of the concept dependency diagram type, let's examine once more the nature of such tools in order to be able to answer questions concerning their utility and usability. At the present time, no exact theories of concept dependency have been developed, although such development seems feasible on the basis of the tools of predicate calculus, in particular using the theory of conceptual graphs [3]. Therefore, it is too early to specify the exact nature, and the degree of details, of the concept relationships one can expect to be reflected in concept dependency diagrams and our above examples are meant as mere illustrations of a basic idea. However, one can formulate some necessary requirements to a tool of this type.

First of all we suppose that we are dealing with clear enough concepts each of which can be -- at least in principle -- explicated in a formal way using a limited number of other concepts and the standard logical tools of predicate calculus. Secondly, we can envisage the three main sources for concept dependency relationships to be included in the diagrams. The first of such sources are the definitions of concepts; we will expect that such definitions make use as defining concepts of all other concepts whose understanding is necessary and sufficient for making accessible the understanding of the meaning of the defined concept. The second type of source for the recognition of concept dependencies are the self-evident semantic relationships between concepts which remain undefined in the frameworks of a given knowledge representation system; such relationships should be explicitly expressed in the form of meaning postulates. The third, but not less important source for concept dependencies, are the newly observed or hypothesized facts whose description involves formerly unrelated concepts; for such concepts an empirical type of dependency has to be established. Special attention should be given to empirical concept dependencies reflecting the emergence of unanticipated usages of formerly known objects for new purposes, i.e., the new application possibilities found or envisaged for formerly known entities, tools or procedures, created independently of such applications.

A general requirement to all dependencies to be reflected in the diagrams is that the ultimate sources for their recognition via the above formally expressed (or expressible) propositions should be specific texts of scientific publications containing the corresponding explicit natural language statements. A concept dependency diagram will reflect a specific stage of development of the represented fields of knowledge.

From this general characterization of concept dependency diagrams, it is clear that at least some partial order will be established among the included concepts. Each definable concept can be characterized as a subordinate to a number of more or less distant undefined (elementary) concepts. The number of elementary ancestors as well as the intermediary superior (simpler) concepts can serve as some measure for the degree of specialization and sophistication of a given concept. The number of concepts which are dependent of a given concept can be interpreted as an indicator of the cognitive importance of the latter. The distance between two concepts and their common ancestors as well as the number and degree of sophistication of common (versus non-common) ancestors could be used for the characterization of some intuitively felt degree of semantic relatedness between concepts [32]. One can also expect that concepts will tend to form clumps made of closely related concepts with a higher semantic distance between the concepts of separate clumps. Speaking in general terms, one can say that the structures represented in concept dependency diagrams will serve to visualize important semantic and cognitive relationships intuitively perceived between concepts as well as such relationships perceived between fields and subfields of knowledge.

One can envisage the possible applications for the concept dependency diagrams.

Firstly, such diagrams can be used to trace some (possible shortest) pathways between a certain initial set of concepts accessible to an individual (user) and some relevant to his interests new concept or new scientific publication. The pathway can be translated in a specific sequence of texts, including, e.g., review articles and/or monographs and selected research publications necessary to be digested in order to achieve a complete understanding of a new key publication or of a new emerging subfield, in particular related to some new applications or new solutions to well-defined problems. On the other hand, the pathway can be used as a guide to a sequence of key concepts which have to be grasped in a more superficial way (for instance in terms of their mere empirical dependencies from other concepts or their applicability for achieving certain goals) in order to reach an approximate understanding of the basics of some new domain. These types of usage essentially related to learning will designate as "cognitive navigation."

Secondly, the diagrams can be used as information retrieval tools for identifying the set of concepts, and through them the set of publications, related to a specific not yet solved theoretical or practical problem. Such a problem is expressed first in terms of one or a few concepts and the diagrams are used to backtrace from them to ancestor concepts corresponding to some other well-developed fields of knowledge whose results are therefore potentially useful for solving the initial problem. This type of usage is in a sense reverse to the previous one insofar as here we move away from a given target concept toward its predecessors in well-developed conceptual domains while the cognitive navigation involves movement from some less specialized known concepts toward a more specialized target concept.

Thirdly, the dynamic of change of conceptual dependency diagrams reflecting the status of knowledge for consecutive periods can be used to detect the most important features of the develoment process of scientific knowledge.

Through the publications, correspondence can be established between concepts and the authors who introduced them as well as the authors who are making essential use of them. This correspondence can be extended further to the social entities, such as laboratories, research institutions and scientific schools in which the authors participate. Therefore, the concept dependency diagrams can be also used for establishing relationships between the conceptual structures of knowledge and the social structure involved in the production of knowledge.

The role of concept dependency diagrams is fulfilled in a fragmentary, and more or less implicit way, by scientific encyclopedias, textbooks, monographs and review publications. The compilation of such tools used for the communication of compressed knowledge is labor and cost intensive. Usually there is a significant time lag between the knowledge status such tools reflectand the real current status of the given knowledge domain. Due to the different time lags for tools covering different comains, they would never convey the correct global picture of the status of knowledge at any given time, even if they would cover in their entirety the whole universe of knowledge.

Such problems could be easier solved if computerized date processing techniques would be made available for bridging the gap between primary scientific publications and the concept maps reflecting the status of knowledge they define. In the next section we will review the main avenues along which progress is being made toward this goal.

Here we will consider, in the abstract, the ideal ways for constructing concept dependency diagrams intended to be used as concept maps of a global scope. We will deliberately disregard all the practicalities of the job and inhabit for a short time a hypothetical ideal world where the metascientific efforts, i.e., resources spent to analyze and digest the results of knowledge acquisition, could equal and even surpass the efforts spent on proper knowledge acquisition.

Even in such a hypothetical utopian world, scientific publications would be written in natural language and not in any logical language. Woodger [33-34] and Carnap [16] have given examples of extensive enough fragments of knowledge taken from biology and physics which have been reformulated using predicate calculus language. So one can suppose in principle that an army of qualified logicians familiar with the corresponding subject domains would translate the basic content of every new scientific publication into a unified logical language. The most important step in doing this would be to identify the "known" concepts the work is using as prerequisites; and this would involve the pinpointing of previous publications where such concepts have been introduced with the same meaning or the detection of differences in the meaning presupposed for the given concept in virtue of its previous usages and the actual meaning this concept is being used in the analyzed paper.

Such activity would result in the essence of the analyzed work being reflected as a formalized record of the deductive reasoning processes which led in this work to proven new conclusions or as a formalized record of new facts observed in its course; such records would make use of the prerequisite concepts and may involve the introduction of new concepts via definitions. In rare cases new undefined concepts would be introduced with a set of meaning postulates; that would signal the appearance of more or less revolutionary works aiming at the revision of the previously used cognitive paradigms.

As a somehow minor side effect of this hypothetical accumulation process of a formalized body of newly created knowledge, a concept inventory would be also created and maintained along with the detection of concept dependency relationships.

So the compilation of concept maps would be the result of an ideal "bottom-up" procedure and the resulting maps would be mere external outlines of the detailed inner content of the formalized body of knowledge.

In a still ideal but computerized world the equivalent of the described analysis procedure, based on intellectual effort, would be an utopian fully-automatized processing flow, resulting in the translation of natural language scientific texts into formalized logical language. The automatic compilation of detailed concept maps would be a spin-off of this process.

The feasibility of both types of the above ideal procedures, even in principle, can be disputed from two related but slightly different points of view. First, it can be argued that the inherent fuzziness of human thinking would lead to the unavoidable failure of any global effort to create in the spirit of a positivistic vision a unified crisp logical language of science built on a unified concept system. Secondly, one can point to the well-recognized by many thinkers properties of polysemy and polymorphism of natural languages, manifested in scientific texts not in a lesser degree than in everyday situations (see Nalimov [35]), which would make impossible the precise understanding of the meaning of scientific texts and therefore prevent their correct rendering in a logical language even if such would be created.

These arguments have serious degrees of theoretical validity and even more practical significance. They case serious doubts upon the possibilities of building detailed and correct in all respects concept maps reflecting concept dependencies in an unequivocal manner.

However, all this does not invalidate the idea of a concept map based on the formally definable dependency relationships between predicates. It means only that the outcome of each individual attempt to build some fragment of such a map involves necessarily an element of probabilistic uncertainty. This will result, in a manner similar to the outcome of any measurement, in a certain degree of variance in the outcome of particular attempts. We can expect though that repeated attempts will lead to the convergence of the results towards an averaged picture of concept relationships. All this is in agreement with the fact that for each given individual the perceived by him specific relationships between some given concepts, as well as the precise meaning of the particular concepts, can differ from one instance to another. Even more differences will be detected between the perceptions of different individuals. Despite all this fuzziness, the system of scientific concepts of a given discipline has at any given period of time a high enough degree of objective existence to make possible the processes of scientific communications and the creative usage of concepts in current scientific research.

In the next section we will review practical ways for obtaining different kinds of approximations to an averaged global map of concept dependencies. These methods are based on automatic data processing techniques and as such they overcome problems concerning the labor intensiveness of an ideal intellectual process. Nevertheless, when one tries to evaluate the results of such automatized procedures, the only effective way is to compare these results with the outcome of particular attempts to build fragments of concept maps for limited domains using intellectual effort. Even better would be the comparison with the fragmentary concept dependency diagrams derived by the intellectual analysis of the content of small sets of specific publications sharing some number of key concepts. The application of formal logical techniques will be important for the consistency of the intellectual analysis.

5. Practical Ways for the Computerized Approximation of Concept Maps.

A. Citation Based Methods.

Most of what comes closest to automatized attempts of building concept maps is based on citation analysis. There are two different reasons for this. The first is that bibliographic citations proved to be the type of records of subject relatedness between scientific publications most easy to use for automatic processing. The second is the successful activity of the Institute for Scientific Information (ISI) in

Philadelphia which led to the creation of a very large database of citations.

Scientific concepts are expressed in natural language texts mostly by groups of words called terms. Terms expressing the same concepts can vary a great deal due to the polymorphy of natural language. Unlike natural language terms, bibliographical descriptions of scientific publications used as references (citations) happen to be much more standardized and whatever further standardization is necessary for them it can be more or less easily accomplished by clerical effort and mechanical manipulations.

According to the basic assumption of citation analysis expressed above in (22) a citation is the indication of the occurrence of a common concept in the citing and cited papers. The nature of the relationship created between the citing and cited papers has been elaborated by H. Small [29,36], who has shown that, at least for the often cited publications, the concept shared by the different citing papers and a given cited paper happens to be one and the same. The verbal expression of that concept can be located in the context of the citing publication in the vicinity of the reference (or reference indication). Thus the citations to a given paper become tags (symbols) for a specific concept, we will further call the "cited concept."

The likelihood for the "cited concept" to be the same for all the instances of citedness is high enough (76-87%) for highly cited papers but it is less than 100%; for more rarely cited papers not enough data are available. However, we can take the uniqueness of the cited concept as basis for a very simple model which can facilitate the understanding of why the transition between citation networks and concept dependency diagrams is possible.

A citation network is a directed graph with nodes corresponding to scientific publications in which the citations are represented by arcs directed from the citing to the cited publications. The database of the Institute for Scientific Information includes around 10^7 journal articles and books from practically all fields of knowledge published from 1955 until the present. The citation network engendered by this number of nodes comprises approximately 1.5×10^8 arcs. Such a huge structured database is beyond the storage and processing capabilities of the present generation of memory devices and computerized database management systems and therefore at any particular moment significant portions of the citation network data are stored on magnetic tapes. Figure 6, reprinted from E. Garfield's book [21], gives a visual image of a very small fragment of ISI's citation network. The fragment consists of 28 nodes and the arcs linking them; the nodes which in the summary network receive a total of 15 or more citations are represented by black circles.

According to our simplified model, each publication can be considered as a tag for a certain "cited concept." Each node of the citation network which is receiving some number of citations, e.g., nodes 1-19 of figure 6, is establishing some kind of link between one of the concepts used in the publication represented by the node (this is the concept which becomes the "cited concept" for this paper when it becomes cited) and the different "cited concepts" of other publications linked to it by incoming citation arcs. In this respect the role of the citing node can be viewed as analogous to the role of "definition"-type nodes (white circles on the concept dependency diagram on figure 2) which create links between a number of "defining" concepts and one "defined" concept. So one can consider a citation network as the immediate equivalent of a simplified concept dependency diagram in which each publication node corresponds to a unique concept, namely to the "cited concept" of the given publication. Verbal concept tags for the concepts represented in such a network could be obtained by applying computational linguistic techniques for processing the corresponding "citing contexts" of the different publications, citing the same publication, in order to detect in such text fragments the common name of the "cited concept" [37]. Such procedures can be easily applied when full texts of publications are available in machine readable form.

Our model is certainly only a poor approximation to the reality, nevertheless, it clearly illustrates the close correspondence between citation networks and concept dependency diagrams and the possibilities of fully automatized transition from the former to the latter.

One way of making citation networks to better conform with the assumptions of our simple model is to consolidate them by omitting the nodes receiving only one or a few citations. Each node of such a consolidated network, consisting only of nodes corresponding to highly cited papers, corresponds with a higher degree of probability to some unique concept and at the same time happens to be cited by a larger number of papers; in other words, in the resulting concept dependency diagram the represented concepts are such that more other concepts depend upon each of them. A concept upon which many other concepts depend is likely to be an important concept. Therefore the consolidation of a citation network not only facilitates its interpretation as concept dependency diagram and makes it more manageable by reducing its size but also achieves a meaningful selection of the represented concepts.

The possibility of using consolidated citation networks as concept dependency diagrams is confirmed by the successful usage of fragments of consolidated citation networks for depicting nodal events in develpment of new areas of research [26,28]. In particular, the citation network of Fig. 6, consisting of highly cited nodes (white nodes are cited, in the overall ISI citation graph of which this fragment is a subgraph not less than 5 times, black nodes more than 15 times) is at the same time a "historiograph" correctly indicating the major advances in genetics between the years 1958 and 1967 [26]. Most of such nodal events or major advances involve the introduction of new concepts.

A concept dependency diagram of the order of magnitude of a global citation network would be practically impossible to use without some meaningful partitioning into regions corresponding to some subdivision of the universe of knowledge into subfields, characterized by dominant key concepts. Most of the development efforts in citation analysis were spent for this purpose.

Computerized techniques successfully used for partitioning sets of objects for which similarity measures are available are known as clustering techniques [38-44]. The size of the sets is critical for their feasibility. Different similarity measures for pairs of nodes of citation networks, derived from their topological interaction with the rest of the nodes of the network have been recently proposed [45]. Such measures are interpretable in terms of similarities of the dependency relationships involving the concepts corresponding to the nodes. However, the size of the global citation network, even if consolidated, remains much too large for the computational applicability of clustering techniques on such scale.

A satisfactory way of overcoming the difficulties related to its size is a piece by piece treatment of the global citation network. A more amenable to treatment single piece is obtained in a most natural way by taking the set of publications which appeared during a given period of time (usually a year) together with all the publications which happen to be cited by them. Such an "annual piece" of the global citation network is still very large: e.g., the "annual piece" for 1983 of the ISI citation network covering only hard sciences and technology consists of around 430,000 nodes representing 1983 publications together with around seven million nodes for publications cited by them; the piece contains around eight million arcs corresponding to the citation links. Such an annual piece is a directed graph consisting of two non-overlapping sets of nodes: citing and cited nodes. A small fragment of such an annual piece of global citation network is shown on Fig. 8.

Let's re-examine, for this type of citation networks, the problem of inferring for these nodes similarity measurers interpretable in terms of relationships between the concepts occurring in the corresponding publications. For this purpose, retaining the model which assumes the uniqueness of the "cited concept" for each cited publication we will further assume that the set of concepts used in any scientific paper consists of two subsets: the subset of prerequisite concepts and the one-element set consisting of the "resulting" concept. The latter is derived from the prerequisite concepts by the observation (or hypothetical consideration) of certain empirical (factual) relationships between some of the prerequisite concepts, followed by a definition type of construction which creates a linkage between the resulting concept and all its prerequisites. The unique "cited concept" is presumably identical to the above "resulting" concept. Each citation is the indication of the citing paper's sharing a prerequisite concept with the cited paper, where the same concept plays the role of resulting concept.

Two different ways of meaningful partitioning the annual citation network into subregions have been investigated. One method achieves first the partitioning of the set of cited nodes, while the second method, at present under development but not yet been tested extensively, will start by partitioning the set of citing nodes.

The first method known as co-citation clustering makes use of a consolidated annual citation network in which only nodes cited more than a certain number of times are included. With threshold values in the 15-17 range, a very significant size reduction of the set of cited nodes is achieved through consolidation: from several million cited nodes only ~ 1-2% are left. The similarity measure for pairs of cited nodes, used for the clustering of these relatively few highly cited nodes is derived from the number of different citing documents which co-cite a given pair of cited nodes. (For the formal definition of co-citation see (3) above; examples are shown on Fig. 8.)

The relationships between the concepts occurring in the publications participating in the co-citation links can be interpreted in the following way from the point of view of our above model. The co-occurrence of two given cited references list of a citing publication means the two "resulting" concepts of the co-cited documents are included in the set of prerequisite concepts of the citing paper. According to our model, we assume that the content of the citing publication consists of a cognitive transition from the prerequisite concepts to the resulting concept. We further assume that this transition consists in part of the observation, detection (or hypothesizing) of empirical links involving certain pairs of prerequisite concepts. A single instance of co-occurrence of two given concepts among the prerequisite concepts of some citing publication cannot be considered as a valid indication of an empirical dependency being created between this pair of cited concepts. When the number of such instances in difference citing publications exceeds a certain threshold value, established in a way to minimize the chance that such co-occurrences might be random events, the multiple co-citation link is a valid indication for the existence -- as perceived by the current science -- of a factual dependency link between the given pair of concepts. Speaking in more general terms, one can say that the similarity of two cited nodes defined using some measure of co-citation strength can be interpreted as the indication of the productivity of the combined usage of the two corresponding concepts in generating new results at the current stage of development of knowledge. Small has examined experimentally the relationships recorded in citing contexts for specific pairs of co-cited concepts and has concluded that strong co-citation corresponds to a high degree of consensus among different citing authors in respect of the relationships they assume between the cited concepts ("the way they connect" these concepts) [37]. He has also sketched a possible classification of such relationships according to types; among the types encountered in his study were such as: "**A** is contained in **B**; **A** and **B** differ in their relationship to **C**; **A** undergoes or is subject to **B**; **A** is used to perform **B**." All these relation types fit the general mold of an empirical (factual relationship between concepts **A** and **B**.

Small has interpreted the established by authors of different citing papers uniformity of the connection between specific pairs of concepts ("prerequisite concepts" according to our model) as an indication to the acceptance by those authors of a certain Kuhnian paradigm. Therefore, it is easy to hypothesize that all those citing papers in which this uniform knowledge pattern was detected through co-citations are expected to belong to a same research specialty. Less obvious, but still reasonable is the assumption that the knowledge pattern characteristic for a specific research speciality can be represented by a connected graph consisting of its key prerequisite concepts linked by those empirical dependency relationships between such key concepts which are prevailing in the current literature. This means specifically that if cited concepts **A** and **B** are connected in the current literature prevailingly by a specific type of relationship, represented by an **A-B** arc of a graph, and if concepts **B** and **C** are similarly connected by some consensual relationship represented by a **B-C** arc of the same graph, then not only publications co-citing **A** and **B**, or **B** and **C**, but also those co-citing **A** and **C** belong to one and the same research speciality dominated by an overall paradigm. The reasonability of this assumption can be supported by arguing that empirical concept dependency relationships are most likely to be transitive. Indeed, the existence of the dependency relationships **A-B** and **B-C** can be interpreted by definition as indicative of some implicit empirical dependency relationship **A-C**, even if such is not expressed explicitly often enough in the current literature. This reasoning can be generalized by induction to show that the dependency relationship will exist between pairs of cited concepts belonging to a same connected component of a graph whose arcs correspond to empirical relationships reflected in the literature. However, when such relationships are detected indirectly via co-citation data, with some degree of probability, the accumulated effect of such probabilities makes the validity of the above reasoning less credible as the distance in the graph between the respective nodes increases.

All the above leads to the conclusion which was anticipated by Small when he successfully applied the single link clustering method to the set of highly cited nodes of consolidated annual pieces of ISI's citation network; he used co-citation links of a strength equal or higher than a fixed threshold value. Small and Griffith [46-47] found that the connected components of the so obtained annual co-citation graph consisting of cited papers called "core papers" correspond well to some subdivision of the universe of knowledge into subfields intuitively perceived as perfectly reasonable by specialists of the corresponding subfields. The "core papers" of a connected component of the annual co-citation graph, corresponding to scientific specialty, taken together with papers citing those core papers is called a "research front."

One of the proofs of the intuitive acceptability of the co-citation clusters as means of valid partitioning of the current scientific literature into subsets of papers corresponding to current research specialties is the fact that specialists could easily ascribe to each such cluster a specific enough name (word phrase) which correctly expressed a general concept characteristic for the given specialty.

Furthermore, the initially obtained co-citation clusters could be subjected recursively to further aggregation procedures due to the fact that the sets of papers citing the core papers of the initially obtained clusters were overlapping and therefore the co-citation between these primary clusters could be used at the second stage of aggregation as the basis for the same procedure of single link clustering; similarly the co-citation between the so-obtained clusters of clusters, or between the aggregates obtained during the two consecutive stages of processing was used for single link clustering at each stage. In this way a four level hierarchy of clusters is being created annually at ISI; at its upper level it comprises less than ten large aggregations, corresponding to the initial subdivision of the universe of knowledge into traditional disciplines, and at each lower level gives the gradual breakdown of such large areas into more and more specialized subfields ending at the lower level with the initial several thousand co-citation clusters [48]. Each aggregate of this hierarchic system is labelled by a concept tag (cluster name) so that the spin-off of this activity is a four level hierarchic tree of scientific concepts presumably reflecting partially the current status of concept dependencies.

Another important development of the application of citation analysis to the study of the structure of scientific knowledge was the implementation of the multidimensional scaling techniques [49-50] for creating two-dimensional visualizations of the internal structure of the primary clusters as well as of the further cumulative aggregates. The matrix of co-citation links reflecting the values of co-citation measures is translated by multidimensional scaling into such a two-dimensional representation of the graph in which the metric closeness between nodes best possibly corresponds to the inverse rank ordering of co-citation strengths. Such maps for the primary clusters proved to be useful for revealing meaningful subaggregates and interrelationships between them which were easily interpretable in terms of the details of the internal structure of a research specialty; the maps obtained for the higher level aggregates provided a visualization for the global knowledge structures revealed by co-citation analysis [48,51]. Recently ISI has applied the whole complex machinery of citation analysis to the 1983 annual piece of its citation network including in it for the first time the social sciences together with the hard sciences and technology.

Another spin-off of this activity was the initiated by E. Garfield compilation by intellectual effort of series of minireviews summarizing the current status of selected research specialities as reflected by current annual clusters. Such minireviews are the basis for "ISI's Atlas of Science" series. Small [1984] has envisioned and tested a computerized or computer-assisted procedure which could produce such minireviews by generating a narrative test from relevant fragments of citing contexts by putting them together in accordance with the co-citation relationships depicted by the corresponding co-citation graph. In the two books published the "ISI's Atlas of Science" series [52-53] the selected cluster maps identifying the highly cited and strongly co-cited documents engendering a research specialty (or "research-front") are accompanied by minireviews and listings of current citing publications. A global regional map showing the higher level aggregation structures of the clusters from the given domain is also given.

From the point of view of the automatic approximation of a global concept dependency diagram most important from the above developments are the goods reasons to believe, as it was argued above, that the primary co-citation graphs display a valid picture of empirical dependencies between key prerequisite concepts of a subfield of knowledge. Small has shown on examples [37] how the nodes of primary co-citation graphs can be labeled with concept names taken from the corresponding citing contexts and how even the arcs can be labelled by verbal expressions empirical concept relationships, taken from co-citing contexts. Assuming the availability of machine-readable full texts of documents and using more or less simple text processing techniques, described for citing statements by O'Connor [54], in combination with more sophisticated computational linguistic techniques, one can envisage the full computerization of the concept tagging process for the core papers of primary clusters. By putting together the detailed local and regional concept dependency diagrams (corresponding to the existing primary co-citation clusters) and arranging them in accordance with interrelationships between primary clusters within their higher level aggregations, one could obtain a global concept dependency diagram structured in a multilevel manner and including at present a total of around 40-50,000 concepts.

Assuming the full validity of the above methods, a still important question to be answered concerns their exhaustiveness. Remarks Small: "The objection can be raised that not all concepts of importance in a field are associated with specific cited documents. This is certainly true for concepts which have become so commonplace that they have assumed the status of tacit knowledge in Polyanyi's sense. Since citation data are time-bound. . .the more accurate question is: Are there important ideas which have never been associated with any cited documents? An answer to this question is much more difficult." [37]

Not less important is to independently confirm the validity of what these methods can deliver and in particular to exclude that part of their results which may prove to be the artifact of the used methods. In the latter respect thorough theoretical considerations of the methods can be certainly useful, as e.g., was the recent study of the statistical validity of the co-citation clustering procedures [55] based on the theory of random graphs. It concluded that between certain threshold values of co-citation strength the results are statistically valid and can be interpreted.

A most desirable way of validation would be completely independent ad-hoc compilation of concept inventories for a number of limited domains followed by the compilation of the corresponding concept dependency diagrams. Such work should be performed by competent field specialists using in some degree symbolic logical tools for content analysis. The results so obtained would be subsequently compared with the results of citation analysis. Such validation studies are extremely labor and cost intensive. More feasible can be studies using alternative data processing procedures and methods for attempting to show the convergency of their results with the results obtained by co-citation based methods. In the next section we will review some relevant word analysis based methods. Here we will briefly discuss alternative citation based methods.

We have defined above in (21) the "subject relatedness" between two texts, in particular scientific publications, by the occurrence in them of a common scientific concept. According to our model regarding the concept composition of a research paper the subject relatedness indicated by a bibliographic reference (see (23)in Section 3) is an indication of the sharing of a concept between the citing paper, where this concept is a "prerequisite" for research, and the cited paper where the same concept is a "resulting" concept of research. Intuitively this can be interpreted as an indication of a certain degree of overlap between the general topic of the two papers.

Another strong kind of subject relatedness will take place when two research publications share their "resulting" concepts, which can be characterized as their main topic; such concepts are likely to be expressed in the "conclusion" portions of papers and to be found in their abstracts and titles. Theoretically the identity of two "resulting" concepts should correspond to the coincidence of their respective "prerequisite" concepts. If one would assume that all the non-tacit prerequisites are unequivocally indicated by identical references, one would conclude that the coincidence of the main topic of two articles can be detected by their identical reference lists. In reality the explicit display through references of the usage of even those concepts which have not yet achieved the status of tacit knowledge is incomplete and subject to many kinds of variations. Therefore the measure of the partial overlap of the reference list of two publications can be expected to be a good indicator for the probability of the coincidence of their respective topics. The complete identity of the resulting concepts of two different pieces of research work may be only a rather rare occurrence. Much more likely the topics of two research papers may have a more or less high degree of similarity. Such similarity of main concepts can be expected to correspond to the similarity of the sets of simpler prerequisite concepts from which such concepts are derived. Since prerequisite concepts are expressed in a high enough degree by cited references, one can expect that the measure of overlap of reference sets of two specific papers one can be a good indicator of the similarity of their main topics.

The existence of a common element between two sets of references was defined (see definition (4) in section 2) as bibliographic coupling; the concept was originally introduced by M.M. Kessler [59-61]. The measure of overlap between such sets, expressed in the number of common elements as such, or appropriately normalized is called the strength of a bibliographic coupling link [62]. Our recent study of the occurrence of bibliocoupling in the entire (non-consolidated) 1983 piece of ISI's citation network has shown that all 1983 publications having non-empty reference lists are bibliocoupled to some other publications; for more than 92% of publications the set of bibliocoupled 1983 papers contained three or more papers.

Selecting for each paper only the three most strongly coupled with it papers we were able to build a network including around 430,000 1983 publications representing the universe of knowledge of science and technology. This network is a directed graph with nodes corresponding to 1983 citing publications. The links between publications, although generated by the symmetric relationship of bibliocoupling, correspond to the assymetric relationship: "publication B is among the three most strongly coupled with publications A." For approximately half of the arrows A-B linking A with its three stronger coupled neighbors there are no corresponding B-A arrows, because the coupling strength of all three strongest coupled with B items exceeds the strength of the coupling link B-A (which is, of course, the same as the strength of the link A-B).

As anticipated, the strong enough bibliocoupling proved to be a good indicator of strong subject relatedness: the preliminary results of expert evaluation indicate that in ~85% of cases adjacent nodes in the coupling network corresponds to publications "well related by subject" [62].

A small fragment of the 1983 bibliocoupling network is shown in Fig. 8, where the nodes are represented by the titles of publications. The subject relatedness is due to the coincidence or to a high degree of similarity of the main concepts of publications as reflect in their titles. Although the concepts expressed in titles are only a rather poor and usually generalized approximation to what can be called the "resulting concept," they are convenient for putting in evidence the semantic relationships between main topics. On Fig. 8 the arrows indicating bibliocoupling are marked with the value of the coupling strength and the verbal expression of the common concept, as inferred from the titles, responsible for the given semantic link. In Fig. 9, these common concepts are made more visible by the introduction of a different type of "concept nodes," so that the coupling links correspond to links through common concepts. Without going into details, one can see that the transformation of this fragment of coupling network into a corresponding concept-document network can be envisaged as an essentially algorithmic procedure. By another algorithmic process, involving the elimination of the document modes, the graph of Fig. 9 was simplified to give the diagram of Fig. 10. This transformation is similar to the simplification procedure by which the concept dependency diagram of Fig. 2 was transformed into the diagram of Fig. 3. The so obtained diagram proves to be a typical concept dependency diagram in which dependencies of analytical genus-species tape appear side-by-side with empirical relationships. We are now in the process of more detailed studies of coupling networks in search of general enough algorithmic procedures for the transformation of such networks into concept dependency diagrams.

A related task is the partitioning of bibliocoupling networks into regions corresponding to subfields of knowledge of the research specialty size. One obvious way we are exploring at present is the application of single link clustering techniques using threshold values for coupling strength. Such studies are potentially important because of the answers the comparison with co-citation clustering can provide to questions concerning the convergency of the results of different citation analysis techniques. On the example of the graph of Fig. 7, one can see that co-citation and bibliocoupling based clustering can give different results.

A different methodology for revealing structural features of citation networks is based on centroid scaling which attempts to place in a multi-dimensional space each citing node in the center of the region occupied by the nodes it is citing and each cited node in the center of nodes citing it [63-65]. This procedure has been applied to small fragments of ISI's citation network and has revealed that the true dimensionality of citation networks is much higher than 2. This method was not yet considered from the point of view of the possibilities to derive concept dependency diagrams using it.

6. Practical Ways for the Computerized Approximation of Concept Maps

B. Word Analysis Based Methods

There have been many attempts in the field of information science to automatically generate word clusters corresponding to groups of semantically related concepts which can be used in information retrieval. Problems which have to be solved before such attempts are performed include the identification, or aggregation, of different forms of a same word and the detection of stable multiword combinations whose components cannot be ascribed to distinct concepts. The similarity measures used for clustering are derived from the frequency of co-occurrence of words in such text units as titles, sentences or whole articles. Recently clustering techniques based on fuzzy sets have been applied for obtaining structures similar to the information retrieval thesauri compiled by intellectual effort [66]. Such thesauri are in essence fragments of concept dependency diagrams and appropriately reflect the dominant dependency relationships between key concepts of usually large enough fields of knowledge. Nevertheless, the results of attempts based on word co-occurrence are consistently different from what can be intuitively interpreted as appropriate concept structures.

Recently a group of French researchers has applied word co-occurrence techniques to the standardized key words intellectually assigned by indexers to scientific papers [67]. As in many previous attempts their work was limited to groups of papers selected in advance as belonging to a same field of research. Taking into account frequency relationships and co-occurrence, this group produced tree-type word structures intended to represent some kind of concept structure existing in a research field. One way of evaluating such structures which they called "leximaps" would be to compare them with the structures created co-citation clustering and multidimensional scaling. Although such comparisons were not yet performed, preliminary inspection of leximaps indicates that the structures reflected by them have much less detail than the co-citation maps. This can be explained by the much higher specificity of citations as concept indicators as compared to words, including most multiword terms and key concepts.

At the same time there are reasons to believe that the word co-occurrence is a rather poor indicator of concept dependency. As wider the knowledge domain, as less precise is the correspondence between word and concept and therefore in wider domains the results of word co-occurrence analysis can be expected to deteriorate. Therefore, a way to improve the ability of word co-occurrence based methods to detect concept dependencies corresponding to concept co-occurrence is to try first to recognize the synonymy and genus-species relationships, and apply the method to groups of words preliminarily created accordingly. We have found in some recent experimental studies that second order association can be useful for that purpose.

This has been anticipated from elementary semantic considerations: synonyms (or near synonyms) are words which one can replace each by the other in a text without changing (or only slightly changing) its meaning. Let's consider the ideally simple case of two-word texts describing one and the same fact (A,B) involving two concepts, A and B, and let's suppose that A can be expressed uniquely by the word a, while B can be represented by two equally frequent synonyms, b_1 and b_2. Our fact (A,B) will be described by texts schematically represented as ($a-b_1$) and, ($a-b_2$). One can see that the synonyms don't co-occur each with other, instead they co-occur with a third word. The word co-occurrence graph for our texts will be $b_1 - a - b_2$, such that the synonyms are separated in it by a path of length 2.

The associations between concepts each of which can be expressed using several synonyms is also easily modeled, with the same final result. For this purpose, let's suppose that in a narrow domain we are dealing with four concepts A, B, C and D and let's assume that there are two empirical relationships between them, (A,B) and (C,D). Each of the above concepts can be expressed by equally frequently used synonyms: a_1, a_2, a_3 for A, b_1, b_2, b_3 for B, c_1, c_2, c_3 for C and d_1, d_2, d_3 for D. If each text describes one fact and the two above relationships are equally frequently recorded, the word co-occurrence graph for such texts is that one given on Fig. 11a. It consists of two connected components, indicating some semantic relationship between the words in each component, but without explicit differentiation between relationships of different kind. Figure 11b shows the graph corresponding to paths of length 2 in the graph of 11a, i.e., it indicates the second order associations. Since only nodes corresponding to synonyms are connected by paths of length 2, the four components of the second-order word co-occurrence graph reveal the four groups of synonyms. After substituting each such connected component by a single node (corresponding to a word cluster obtained by second order associations) and after representing the linkages between the nodes of the original co-occurrence graph as links between the cluster-nodes to which they now belong (this corresponds to a clustering of clusters) we obtain the graph 11c which correctly shows and differentiates the analytical synonymy relationships between words, from the synthetic empirical relationships.

This is not to assert that second order association graphs can be a general solution for a more proper structuring of the primary word co-occurrence graphs, but only to emphasize that more sophisticated procedures are needed than presently used. The former can require significant computational resources when one deals with very large text collections, e.g., the 5.10^5 titles of publications covered in a year in the Science Citation Index.

An especially difficult problem for the transition from word structures to concept dependency structures is caused by the presence in scientific texts of many homonyms, with such words as "plasma" and "cells" the most prominent examples. Their devastating effect can be illustrated on the above model; it is enough for just one of the twelve words of our example to be a homonym, e.g., $b_2 = d_1$ in order to lead to the failure of the above described procedure. The solution can be the narrowing down of the subject domain of the texts to be processed. One way we found in our current studies useful for this purpose is the preliminary selection of texts (titles of scientific publications) by selecting as "subfield delimitor" a highly specialized term (usually a multi-word phrase), occurring between 10 to 100 times a year, which is unlikely to be homonymic by itself. Such a term, occurring in current publications, delimits a most homogeneous knowledge domain. The limited size of the so obtained subfile and its topical homogeneity make possible a much more "in depth" processing of word structures, including the detection of syntactic structures, which afterwards can be used instead of single words as units subjected to further co-occurrence and clustering type of analysis. Useful for the fully automatic detection of syntactical structures of word phrase type can be the domain independent approximate simple procedures of syntactic-semantical analysis we developed earlier in connection with the algorithmic compilation of subject indexes of key word phrase type [68-69]. As a result of the topical homogeneity of so defined text subfiles, one finds that for pairs of phrases sharing some lexical unit a quasi-synonymy (hyponymy) relationship can be established. When second degree associations are detected between such quasi-synonymic chunks hyponymy relationships are revealed for phrases which don't share any lexical elements. In this way structures consisting of word phrases can be created which presumably reflect important for the sub-domain concept dependency relationships.

In order to build in this way more or less global concept structures, it would be necessary to partition the macrofile of texts into topically homogeneous local chunks, and then to integrate the detected local mini-structures into a global structure. Although such a two (or multi-level) procedure may prove to be computationally feasible, it is doubtful that enough "subfield delimitor terms" could be found to ensure the exhaustive partitioning of the universal file. It seems much more reasonable to use some independent (from lexical tools) methods for achieving such partitioning. This brings us to the consideration of integrated ways of processing which would combine the usage of different types of structures (e.g., citation based and word based) algorithmically detectable in scientific texts.

Before that, we want to briefly mention two more word based methods explored for building structural representations of research subfield. The similarity of sets of key words extracted from texts of abstracts of research papers was used to build groups of papers dealing with homogeneous research topics [70] and for each such group some characteristic subset of key words was inferred which could be considered as a description of the conceptual inventory of the group. The lexical similarity links of strength lower than the threshold values used for primary grouping can be used for linking the groups in an integrated structure which would also correspond to a concept structure. However, since the primary grouping is performed on the basis of lexical similarities of texts, it is most likely the approach will fail if applied to broader fields where the homonymy plays a noticeable role.

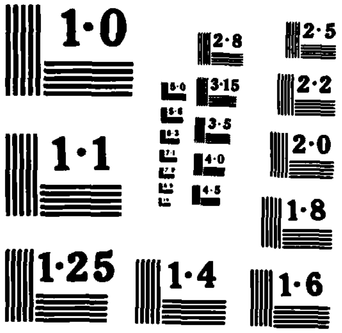
Another method is based on term occurrence detected in the very specific type of **text of captions** for tables or graphical displays of numerical dependencies studied in certain subfields of medicine [71]. The processing of concept pairs so detected was used for building tree-like concept structures.

Summarizing the evaluation of word based methods for revealing concept structures by processing of selected text elements of scientific publications we have to admit that despite several different promising approaches these methods did not yield as yet practical results for the approximation of concept maps.

7. Practical Ways for the Computerized Approximation of Concept Maps

C. Integrative methods

The concept maps we are trying to compile automatically for the approximation of concept dependency diagrams involve necessarily the ultimate labeling of their nodes by natural language expressions, e.g., by multi-word terms or word phrases. Ideally each link of such diagram has to be tagged also by references to the texts of publications, or more precisely, by references to specific fragments of publications



NATIONAL BUREAU OF STANDARDS
MICROCOPY RESOLUTION TEST CHART

where that link is created either as a definition or as the description of a new empirical relationship. So ultimately each concept node of the diagram will be tagged by references to a group of papers in which the concept is defined, elaborated or used.

Since purely word based methods per se are unlikely to be able to generate global concept structures any practical solution has to involve the combination of some different type of method with word based methods. At present, practically significant results have been obtained with co-citation clustering procedures used in combination with the intellectual tagging by word phrases ("research front specialty names") of generic concepts corresponding to clusters.

Methods to automatize such cluster naming procedures on the basis of the extraction of key word phrases from the titles of publication groups created by clustering and the selection of a representative subset of such phrases have been designed at ISI by I. Sher. A problem for such type of methods is the difficulty of detecting the lexical elements corresponding to some generalized concepts potentially useful for naming the cluster, but not occurring implicitly in the titles. H. Small, of ISI, has envisaged procedures for the concept tagging of individual papers on the basis of the automatic processing of citing text fragments taken from citing publications.

The fully computerized process of building reasonable approximations to global concept structures from text elements of scientific publications can be outlined as including the following necessary stages, performed not necessarily in this order. A concept inventory has to be built and this presupposes the detection of synonymy relationships between terms used for expressing this concept in the current literature. Secondly, linkages have to be detected between the concepts, and possibly differentiated as corresponding to relationships of definition type or empirical type. Thirdly, some meaningful, possible multi-level partitioning of the global diagram into regions has to be achieved in order to make it accessible to the user.

In the present co-citation based strategy used for the mapping of science, these procedures are performed more or less in inverse order. When using bibliographic coupling first a network type of structure is established between research papers; the next envisaged step, as illustrated by Figs. 8-10, is supposed to be the compilation of local inventories of concepts for narrow subregions of the network with the simultaneous recognition of quasi-synonymy type relationships between word phrases. The subregions of the network to which such procedures are applied are thought to be of the "sliding window type" of a given radius (=2 or 3) with each node becoming consecutively the center of the window. The processing of a region is performed if the average coupling strength for this portion of the network is exceding a minimal value (~ 2).

Within each region semantically valid links can be established between word phrases sharing at least one lexical element, even if such phrases don't belong to directly coupled titles. For example, for the phrases of the titles of Fig. 8 in this way the following linkage chain can be established between some of the phrases: "motor characteristic -- drug induced ocular motor disorders -- ocular nystagmus." This chain contains the definitional link "nystgmus -- ocular motor disorder," which cannot be detected on the basis of shared lexical elements.

Within the narrow domain corresponding to a tightly coupled region semantic relationships can be hypothesized also between word phrases of the corresponding titles even if they don't share lexical elements. For instance, for the fragment of bibliocoupling netword shown on Fig. 8, semantic relationships can be tentatively established for all the phrases pairs of the following set: "Methods of treatment," "visual acuity," "motor characteristics," "differential diagnosis and mechanisms," "contact lenses," "identical twins," "base-out prisms," "cerebral ataxia." All such tentatively established in a region relationships can be further validated or discarded depending on their repeated occurrence inside a larger piece of network, e.g., including for each given region the immedicately neighboring with it regions. In this way, due to piecewise regional processing, problems caused by homonymy may be avoided.

The described hypothetical procedure illustrates a certain degree of interactive integration between word co-occurrence and citation based methods. The final result is supposed to be the transformation of the initial coupling network into a concept dependency network with the retention of the correspondence between its conceptual nodes and the original texts of titles (and, through them, publications) from which they were derived.

Insofar as the partitioning of such global conceptual network is concerned it may prove reasonable not to try to generate some unique preferred partition, but to provide some alternative ones, centered on certain key concepts detected by their key positions (multiple links) in the network and limited in size by some reasonable distance from the key concept(s) of the included concept nodes. Instead of using predetermined partitionings, a conceptual network could be embodied in an online accessible database with the possibility of freely browsing through it; such "cognitive navigation," from any given concept node in any desired direction with the possibility of backtracking could provide to the user the equivalent of a "customized partitioning."

Whatever the success of the procedures envisaged above may be it is important to realize that citations and words are only parts of the elements of the huge existing bibliographic databases which can be used for the detection of concept structures. Some other elements which could be used for the same purposes are author names, journal titles, institutional or corporate affiliations, country and language of publication. For achieving full interactive and integrative processing of all types of available data elements networks have to be constructed in which there are as many different types of nodes as many different data element types, e.g., publication nodes, word nodes, author nodes, journal nodes, etc. In such integrative networks citation based links are established between publication nodes and additional links are created between all other types of nodes and publications nodes on the basis of the occurrence of the corresponding data elements in the bibliographic descriptions of publications. Recently the creation of such data structures was discussed for information retrieval purposes [72]. In the complex resulting network new links between elements of the same type can be created, e.g., quasi-synonymy relationships between words on the basis of existing links (e.g., citation links) between adjacent nodes of another type; some nodes (for instance word nodes corresponding to homonyms or author nodes corresponding to different persons) can be replaced by duplicate nodes in order to discard "false" links created by them between publication nodes which are not linked in any other way (e.g., lack of citation links and journal links).

The truly integrative processing of such complex networks would require many iterations and significant computational power. Even more resources will require the replacement in such networks of the title words of publications, which are rather poor sources for currently used scientific concepts, by words, phrases or sentences selected from abstracts or appropriate fragments (discussion, results, conclusion) of full texts.

The results of such analysis in the form of tentative concept dependency networks will have to be edited and corrected with the participation of many domain specialists. All this may become possible only on the basis of a large scale sustained effort on the basis of the cooperative interaction of representatives of different specialists using the experience of scientographers and the methods of computational linguistics, knowledge engineering and symbolic logic.

The final result may be well worth the effort. The explicit global structural representation of scientific concepts, dynamically updated and tied to the entire growing body of recorded scientific knowledge could be an important step in the creation of mega-expert systems which may really deserve the name of the world brain.

Conclusion

The creation of automatized intelligent systems which could contribute to significantly increase the overall productivity of scientific research requires knowledge representation tools of global scale. As an important step toward the detailed logical analysis and representation of large bodies of scientific knowledge, the automatic compilation from existing databases of an intermediary type of knowledge representation is proposed in the form of a detailed concept dependency diagram of global scope. The relationships are outlined between concept dependency diagrams and detailed knowledge representations based on predicate calculus language and full formalization.

The successful efforts aimed at the representation of the structure of scientific fields, mainly on the basis of citation data are reviewed. The prospects for the development of related methods for the computerized compilation of regional and global concept maps in the form of concept dependency diagrams are discussed.

References

1. J.R. Rumble, V.E. Hampel, editors, "Database management in science and technology." Elsevier, New York, 1984.
2. J. Hayes-Roth, D.A. Waterman, D.B. Lenat, editors, "Building expert systems." Addison-Wesley, Reading, Massachusetts, 1983.
3. J.R. Sowa, "Conceptual structures. Information processing in mind and machine." Addison-Wesley, Reading, Massachusetts, 1983.

4. G. McCalla, N. Cercone, guest editors, "Knowledge representation," (special issue). Computer, 1983, vol. 16, no. 16, pp. 12-122.

5. R. Brachman, B. Smith, editors, "Special issue on knowledge representation." SIGART Newsletter, 1980, no. 70.

6. V.V. Nalimov, Z.M. Mulchenko, "Naukometriya" (Scientometrics). Nauka, Mowcow, 1969.

7. H.G. Wells, "World brain." Doubleday, Doran, Garden City, New York, 1938.

8. E. Garfield, "Towards the world brain," (editorial) Current Contents, 1964, no. 7, pp. 4-5, reprinted in: E. Garfield, "Essays of an information scientist," vol. 1, ISI Press, Philadelphia, Pennsylvania, 1977.

9. E. Garfield, "'World brain' or 'Memex' mechanical and intellectual requirements for universal bibliographic control," in: E.B. Montgomery, editor, "The foundations of access to knowledge." Syracuse University Press, Syracuse, New York, 1968, pp. 169-196.

10. E. Garfield, "The world brain as seen by an information entrepreneur," in: "Information for action: from knowledge to wisdom." Academic Press, New York, 1975, pp. 155-160.

11. O. Neurath, R. Carnap, C.W. Morris, "International encyclopedia of unified science." University of Chicago Press, vol. 1, 1938, vol. 2, 1939, Chicago, Illinois.

12. M. Kochen, "WISE, a world information synthesis and encyclopedia." Journal of Documentation, 1972, vol. 28, no. 4, pp. 322-343.

13. M. Kochen, "Document retrieval based on computer-aided comprehension," in C. Keren and L. Perlmutter, editors, "The Application of

Mini- and Micro-Computers in Information, Documentation and Libraries," North-Holland, Amsterdam, 1983. Keynote Address.

14. R. Schank, P.G. Childers, "The cognitive computer." Addison-Wesley, Reading, Massachusetts, 1984.

15. M. Minsky, "A framework for representing knowledge," in "The Psychology of Computer Vision." McGraw-Hill, New York: P.H. Winston, editor, 1975, pp. 211-277.

16. R. Carnap, "Introduction to symbolic logic and its applications." Dover, New York, 1958.

17. D.G. Bobrow, editor, "Special issue on non-monotonic logic." Artificial Intelligence, 1980, vol. 13, no. 1-2.

18. Non-monotonic reasoning workshop sponsored by the American Association for Artificial Intelligence, October 17-19, 1984, Mohawk Mountain House, New Paltz, New York (abstracts of papers).

19. P.J. Hayes, "The logic of frames," in: D. Metzing, editor, "Frame conceptions and text understanding," Walter de Gruyter, Berlin, 1979, pp. 46-61.

20. J.R. Hobbs, in [5], Chapter 3 (Individual position statements). pp. 48-49.

21. R. Kowalski, ibidem, p. 49.

22. N.V. Findler, editor, "Associative networks: representation and use of knowledge by computers." Academic Press, New York, 1979.

23. G.D. Ritchie, F.K. Hanna, "Semantic networks--a general definition and a survey," Information Technology: Research and Development, 1983, no. 2, pp. 187-231.

24. R.J. Brachman, "What IS-A and isn't: an analysis of taxonomic links in semantic networks," in [4], pp. 30-36.

25. J. Barwise, J. Perry, "Situations and attitudes," MIT Press, Cambridge, Massachusetts, 1983.

26. E. Garfield, "Citation indexing: its theory and application in science, technology and humanities." John Wiley and Sons, New York, 1979.

27. Science citation Index, Institute for Scientific Information, 1964-.

28. E. Garfield, "Citation indexing, historio-bibliography, and the sociology of science," in: K.E. Davis, W.D. Sweeney, editors, "Proceedings of the third international congress of medical librarianship, Amsterdam, 5-9 May 1969," Excerpta Medica, Amsterdam, 1970, pp. 187-204.

29. H.G. Small, "Cited documents as concept symbols," Social Studies of Science, 1978, vol. 8, pp. 327-340.

30. W.V. Quine, "Two dogmas of empiricism," Philosophical Review, 1951, vol. 60, pp. 20-43, reprinted in F. Zabeeh, E.D. Klemke, A. Jacobson, "Readings in semantics," University of Illinois Press, Urbana, Illinois, 1974, pp. 584-610.

31. R.C. Schank, B.L. Nash-Webber editors, "Theoretical issues in natural language processing," Association for Computational Linguistics, 1975.

32. M.R. Quillian, "Semantic memory," Report AD-641671, Clearinghouse for Federal Scientific and Technical Information. Abridged version in: M. Minsky, editor, "Semantic information processing," MIT Press, Cambridge, Massachusetts, 1968, pp. 227-270.

33. J. Woodger, "The axiomatic method in Biology," Cambridge, 1937.

34. J. Woodger, "The technique of theory construction" in [11], vol. 2, no. 5, 1939.

35. V.V. Nalimov, "In the labyrinths of language: amathematician's journey," ISI Press, Philadelphia, Pennsylvania, 1981.

36. H. Small, E. Greenlee, "Citation content analysis of a co-citation cluster: recombinant-DNA," Scientometrics, 1980, vol. 4, no. 2, pp. 277-301.

37. H. Small, "Co-citation content analysis and the structure of paradigms," Journal of Documentation, 1980, pp. 183-196.

38. A.J. Cole, "Numerical taxonomy," Academic Press, New York, 1969.

39. M.R. Anderberg, "Cluster analysis for applications," Academic Press, New York, 1973.

40. B. Everitt, "Cluster analysis," Heineman Education Book, London, 1974.

41. J.A. Hartigan, "Clustering algorithms," Wiley and Sons, New York, 1975.

42. H.J. Clifford, "An introduction to numerical classification," Academic Press, New York, 1975.

43. L. Fisher, J.W. Van Ness, "Admissible clustering procedures," Biometrics, 1971, vol. 58, pp. 91-104.

44. R. Sibron, "Order invariant methods for date analysis," Journal of the Royal Statistical Society, Series B, 1972, vol. 34, no. 3, pp. 311-349.

45. M. Nowakowska, "Theories of research," vol. 1, Intersystems Publications, Seaside,California, 1984.

46. H. Small, B.C. Griffith, "The structure of scientific literatures. I: Identifying and graphing specialties," Science Studies, 1974, vol. 4, pp. 17-40.

47. B.C. Griffith, H.S. Small, J.A. Stonehill, S. Dey, "The structure of scientific literatures." II. Toward a macro- and micro-structure for science," Science Studies, 1974, vol. 4, pp. 339-365.

48. H. Small, E. Sweeney, E. Greenlee, "Clustering the Science Citation Index using co-citations: II. Mapping science," Scientometrics, 1985, in press.

49. J.B. Kruskal, "Multidimensional scaling by optimizing goodness-of-fit to a non-metric hypothesis," Psychometrika, 1964, vol. 29, pp. 1-37.

50. R.N. Shepard, "Multidimensional scaling, tree-fitting and clustering," Science, 1980. vol. 210, pp. 390-398.

51. E. Garfield, M.V. Malin, H.G. Small, "Citation data as science indicators," in: Y. Elkana, editor, "Toward a metric of science: The advent of science indicators," Wiley and Sons, New York, 1978, pp. 179-207.

52. E. Garfield, editor, "ISI atlas of science: Biochemistry and molecular biology," Institute for Scientific Information, Philadelphia, 1982.

53. E. Garfield, editor, "ISI atlas of science: Biotechnology and molecular genetics," Institute for Scientific Information, Philadelphia, 1984.

54. J. O'Connor, "Citing statements: Recognition by computer and use to improve retrieval," Proceedings of the American Society for Information Science, 1980, vol. 17, pp. 177-179.

55. W.W. Shaw, Jr., "Critical thresholds in co-citation graphs," Journal of the American Society for Information Science, 1985, vol. 36, no. 1, pp. 38-43.

56. E. Garfield, "A united index to science," Proceedings of the International Conference on Scientific Information, 1958; National Academy of Sciences, National Research Council, Washington, D.C., 1959, pp. 461-474.

57. H.G. Small, "Co-citation in the scientific literataure: A new measure of the relationship between two documents," Journal of the American Society for Information Science, 1973, vol. 24, no. 4, pp. 265-269.

58. I.V. Marshakova, "A system of links between documents, built on the basis of citations (using the Science Citation Index)," (in Russian), Nauchno-tekhnicheskaya Informatsiya, series 2, 1973, no. 6, pp. 3-8.

59. M.M. Kessler, "Bibliographic coupling between scientific papers," American Documentation, 1963, vol. 14, pp. 10-25.

60. M.M. Kessler, "Bibliographic coupling extended in time: Ten case histories," Information Storage and Retrieval, 1963, vol. 1, pp. 169-187.

61. M.M. Kessler, "Comparison of results of bibliographic coupling and analytic subject indexing," American Documentation, 1963, vol. 16, pp. 223-233.

62. G. Vladutz, J. Cook, "Bibliographic coupling and subject relatedness," Proceedings of the American Society for Information Science, 1984, vol. 21, pp. 204-207.

63. E. Noma, "The simultaneous scaling of cited and citing articles in a common space," Scientometrics, 1982, vol. 4, pp. 205-231.

64. E. Noma, "Untangling citation networks," Information Processing and Management," 1982, vol. 18, p. 43-53.

65. E. Noma, "Co-citation analysis and the invisible college," Journal of the American Society for Information Science, 1984, vol. 35, no. 1, p. 29-33.

66. S. Miyamoto, T. Miyake, K. Nakayama, "Generation of a pseudothesaurus for information retrieval based on concurrences and fuzzy set operations," IEEE Transactions on Systems, Man, and Cybernetics, 1983, vol. SMC-13, no. 1, pp. 62-70.

67. M. Callon, J.-P. Courtial, W.A. Turner, S. Bauin, "From translation to problemataic network: An introduction to co-word analysis," Social Science Information, 1983, vol. 22, no. 2, pp. 191-235.

68. G. Vladutz, E. Garfield, "KWIPSI--An algorithmically derived key word/phrase subject index," Proceedings of the American Society for Information Science, 1979, vol. 16, pp. 236-245.

69. G. Vladutz, "Natural language text segmentation techniques applied to the automatic compilation of printed subject indexes and for online database access," Proceedings of the Conference on Aplied Natural Language Processing sponsored by the Association for Computational Linguistics and the Naval Research Laboratory, 1983, Santa Monica, California.

70. I.V. Marshakova, "The classification of documents on the basis of lexical composition (using documents' key words), (in Russian), Nauchno-Teknicheskaya Informatsiya, series 2, 1974, no. 5, pp. 3-10.

71. J.M. Weiner, Stephen M. Stowe, R.C. Honour, C.D. Hammond, "Assessing scientific performance," Information Processing and Management, 1984, vol. 20, no. 5/6, pp. 575-582.

72. W.B. Croft, R. Wolf, R. Thompson, "A network organization used for document retrieval," Proceedings of the sixth annual international ACM SIGIR conference on research and development in information retrieval, 1983, vol. 17, no. 4, pp. 178-188.

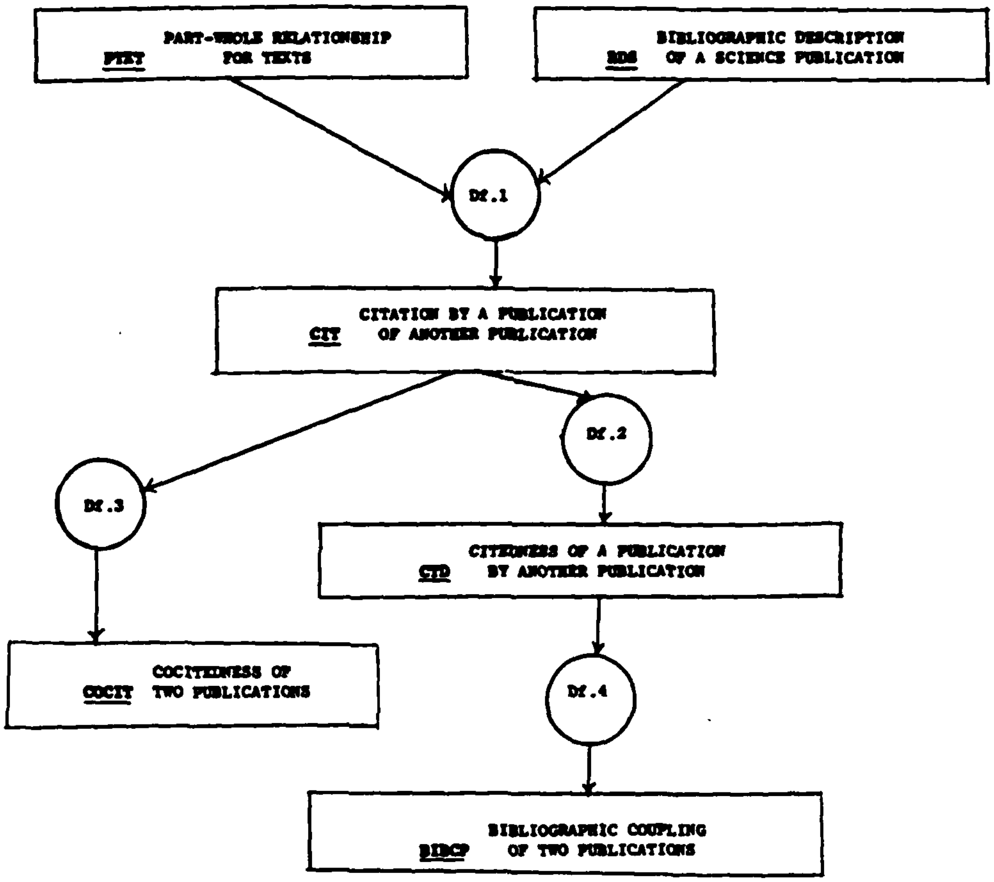

PART-WHOLE RELATIONSHIP
PTXT FOR TEXTS
BIBLIOGRAPHIC DESCRIPTION
BDS OF A SCIENCE PUBLICATION
Df.1
CITATION BY A PUBLICATION
CIT OF ANOTHER PUBLICATION
Df.2
Df.3
CITEDNESS OF A PUBLICATION
CTD BY ANOTHER PUBLICATION
COCITEDNESS OF
COCIT TWO PUBLICATIONS
Df.4
BIBLIOGRAPHIC COUPLING
BIBCP OF TWO PUBLICATIONS

Fig. 1 Graph of dependencies between predicates (concepts) established by a set of definitions

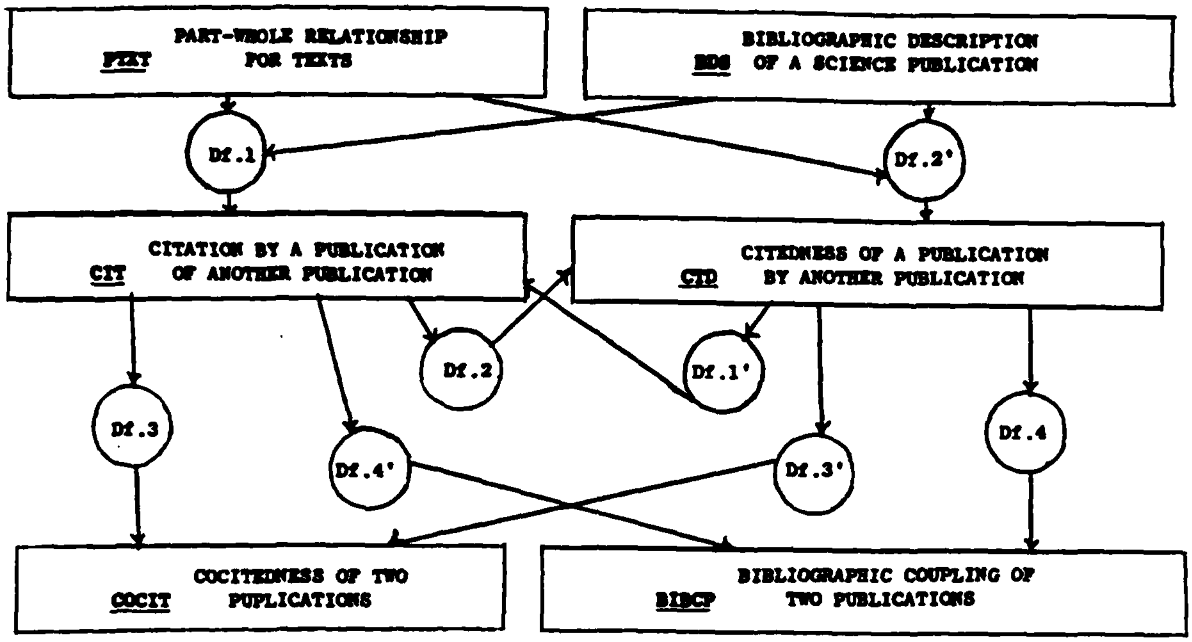

PART-WHOLE RELATIONSHIP
PWRT FOR TEXTS
BIBLIOGRAPHIC DESCRIPTION
BDS OF A SCIENCE PUBLICATION
Df.1
Df.2'
CITATION BY A PUBLICATION
CIT OF ANOTHER PUBLICATION
CITEDNESS OF A PUBLICATION
CTD BY ANOTHER PUBLICATION
Df.2
Df.1'
Df.3
Df.4'
Df.3'
Df.4
COCITEDNESS OF TWO
COCIT PUPLICATIONS
BIBLIOGRAPHIC COUPLING OF
BIBCP TWO PUBLICATIONS

Fig. 2

Graph of dependencies between predicates (concepts) established by definability relationships

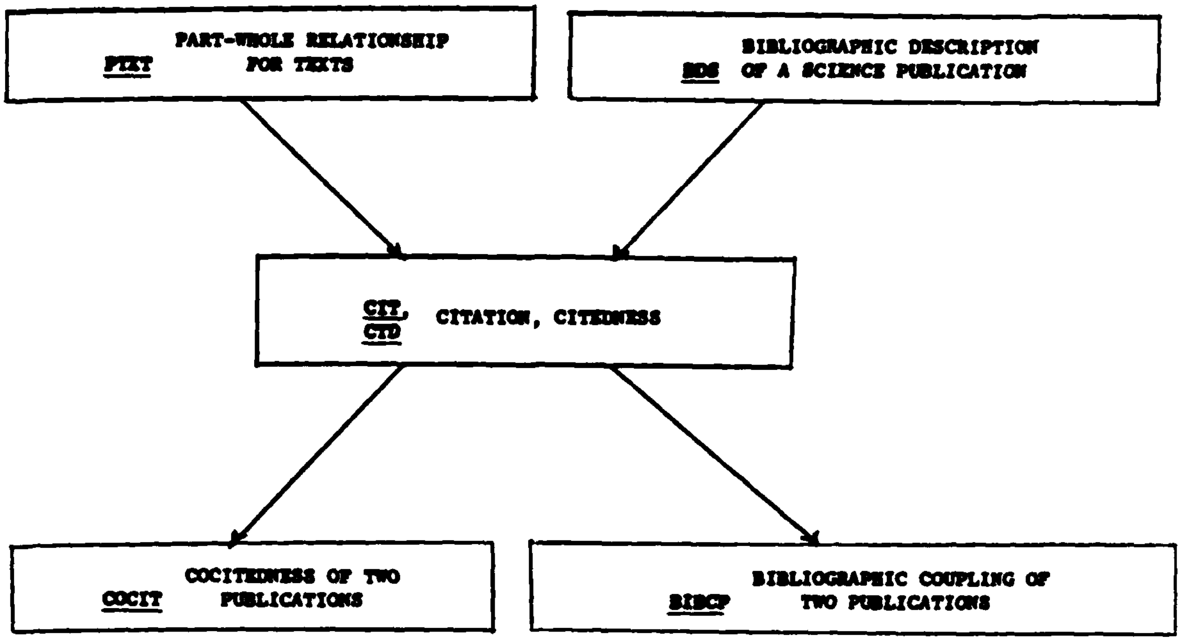


Fig. 3

Simplified graph of dependencies between predicates (concepts) established by definability relationships

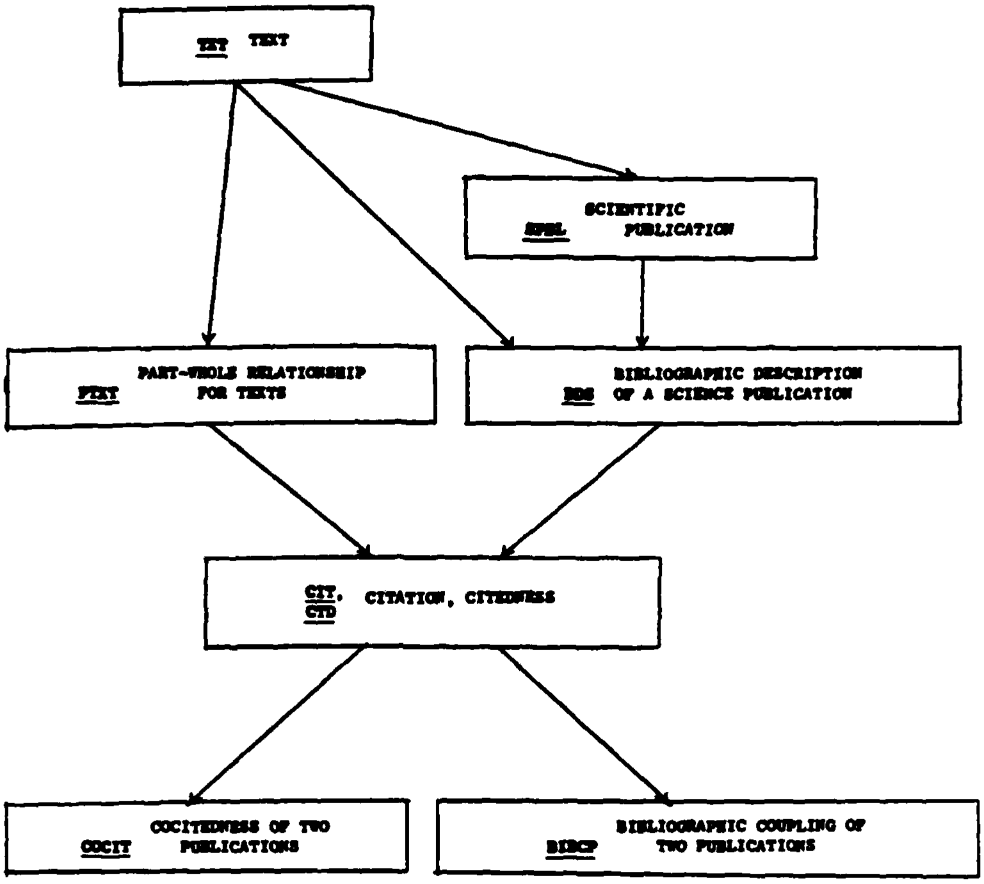
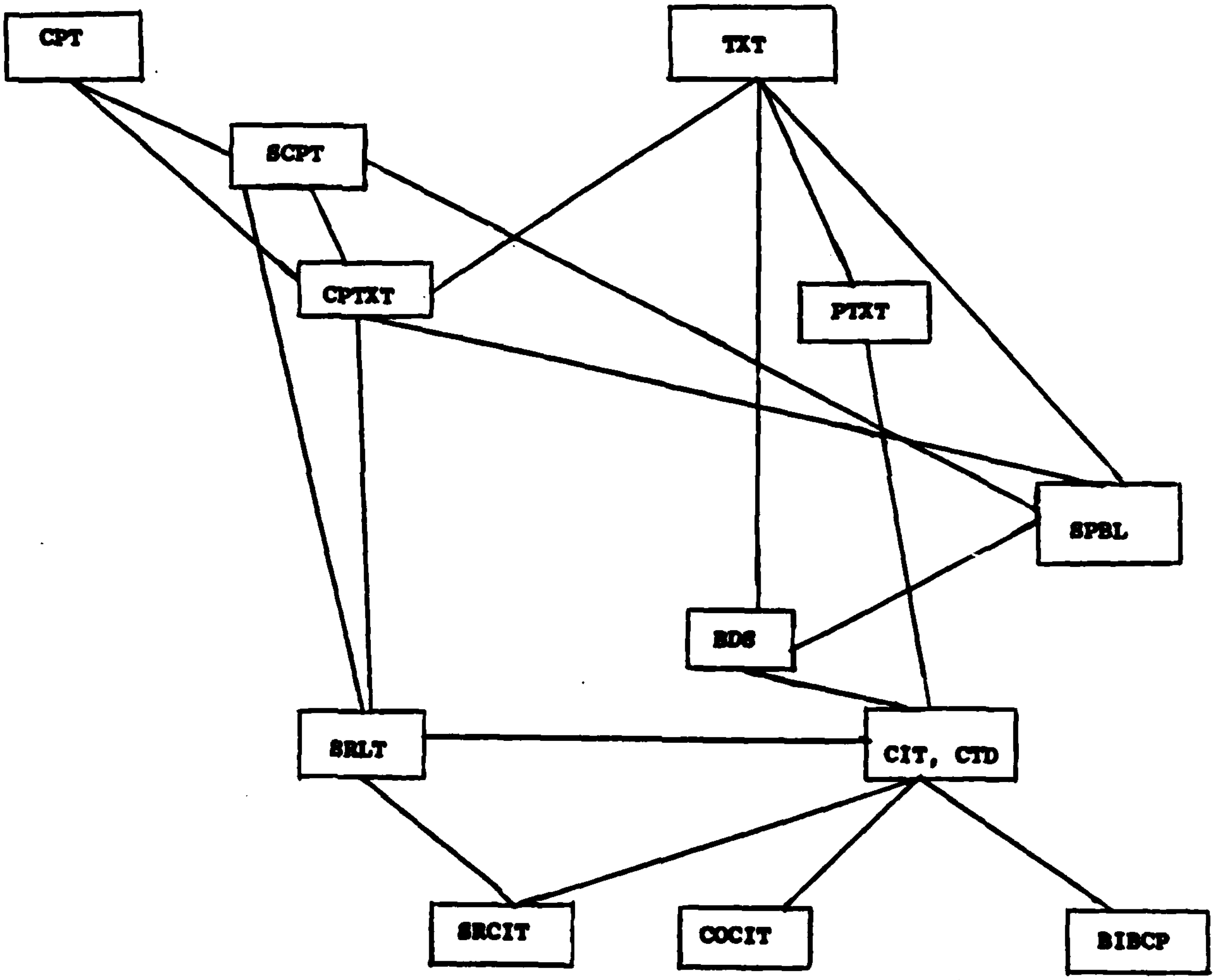


Fig. 4 Graph of concept dependencies established by definability relationships and meaning postulates



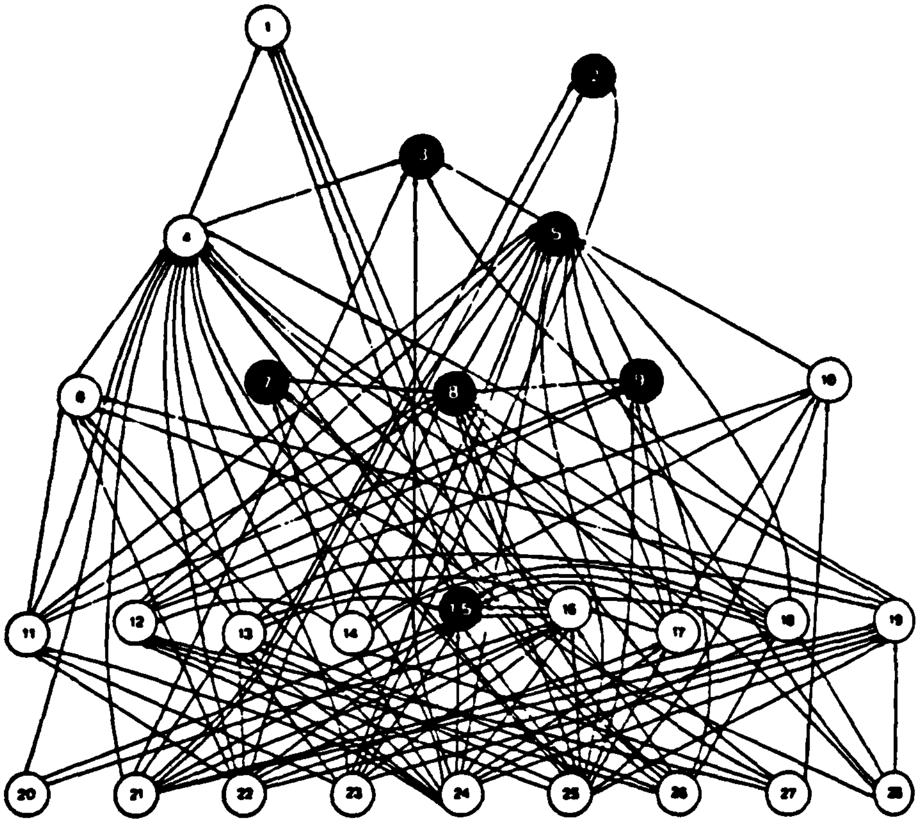
E. Garfield 1958-69 [28, 56]

H.S. Small 1973 [57]
I.V. Marshakova 1973 [58]

M.M. Kessler 1963 [59]

Fig. 5

Graph of concept dependencies established by definability, meaning postulates and empirical (factual) relationships. For the lower row of concepts the authors who introduced them are indicated with the corresponding references (see page 61); The rest of simpler concepts are in the domain of 'tacit knowledge'.



1, Sheehan 1958; 2, Bray 1960; 3, Nirenberg 1961; 4, Marcker 1964; 5, Nirenberg 1964; 6, Marcker 1965; 7, Brenner 1965; 8, Khorana 1965; 9, Nirenberg 1965; 10, Khorana 1965; 11, Marcker 1966; 12, Khorana 1966; 13, Marcker 1966; 14, Khorana 1966; 15, Adams 1966; 16, Webster 1966; 17, Nirenberg 1966; 18, Ochoa 1966; 19, Nakamoto 1966; 20, Berberich 1967; 21, Lucas-Leonard 1967; 22, Caskey 1967; 23, Ochoa 1967; 24, Khorana 1967; 25, Nirenberg 1967; 26, Ochoa 1967; 27, Khorana 1967; 28, Ochoa 1967.

Figure 6 Historiograph of the major advances in genetics between 1958 and 1967, based on a citation analysis of a review of the 1967 literature. Each circle represents a paper cited five or more times by the papers listed in the bibliography of the review. The papers represented by solid black circles were cited 15 times or more in the 1967 *SCI.*

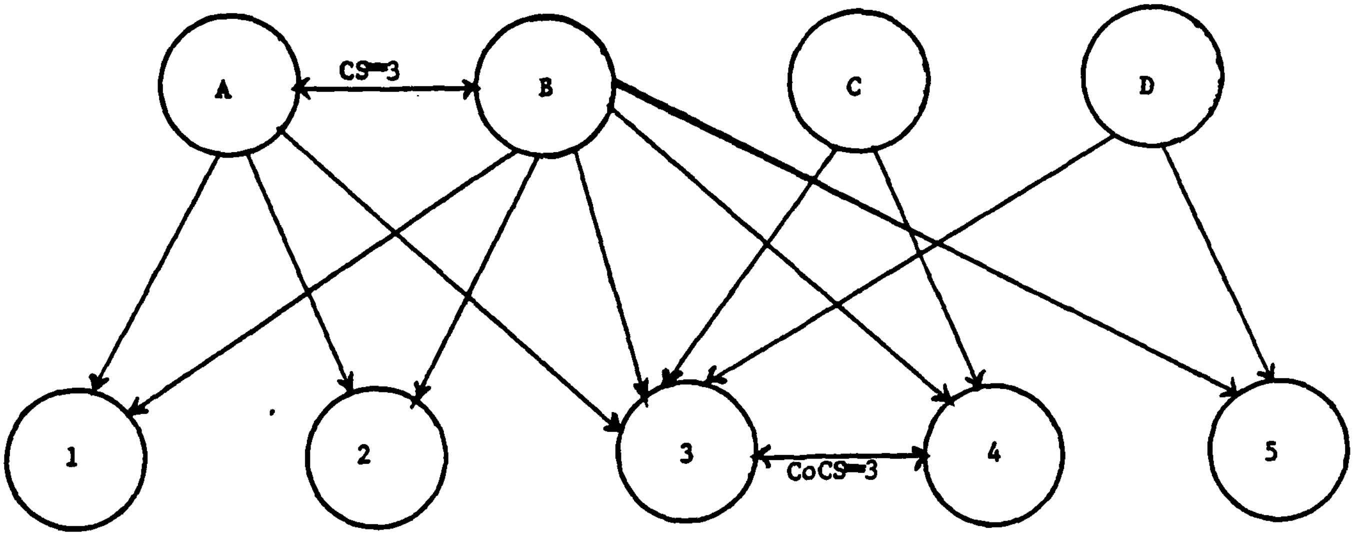


Fig. 7

Fragment of an annual piece of the citation network. Nodes A-D represent cited publications. Nodes 3 and 4 are co-cited by nodes B, C and D; the 3-4 co-citation strength CoCS=3. Nodes A and B share references 1, 2, and 3; the A-B coupling strength CS=3.

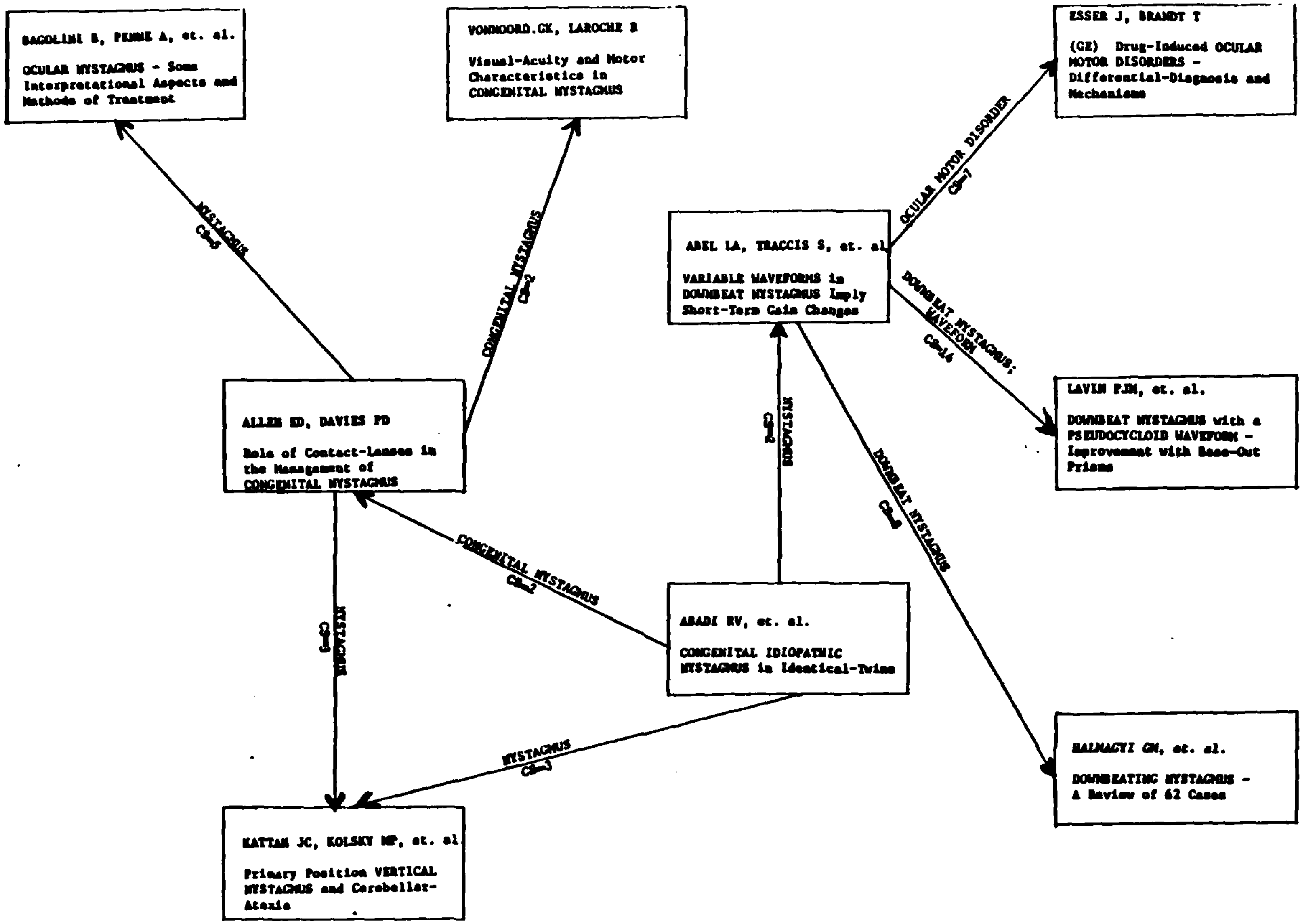

BAGOLINI B, PENNE A, et. al.
OCULAR NYSTAGMUS - Some Interpretational Aspects and Methods of Treatment
VONNOORD.CK, LAROCHE R
Visual-Acuity and Motor Characteristics in CONGENITAL NYSTAGMUS
ESSER J, BRANDT T
(GE) Drug-Induced OCULAR MOTOR DISORDERS - Differential-Diagnosis and Mechanisms
ABEL LA, TRACCIS S, et. al
VARIABLE WAVEFORMS in DOWNBEAT NYSTAGMUS Imply Short-Term Gain Changes
LAVIN P.JM, et. al.
DOWNBEAT NYSTAGMUS with a PSEUDOCYCLOID WAVEFORM - Improvement with Base-Out Prisms
ALLEN ED, DAVIES PD
Role of Contact-Lenses in the Management of CONGENITAL NYSTAGMUS
ABADI RV, et. al.
CONGENITAL IDIOPATHIC NYSTAGMUS in Identical-Twins
HALMAGYI GM, et. al.
DOWNBEATING NYSTAGMUS - A Review of 62 Cases
KATTAH JC, KOLSKY MP, et. al
Primary Position VERTICAL NYSTAGMUS and Cerebellar-Ataxia
NYSTAGMUS CS=5
CONGENITAL NYSTAGMUS CS=2
OCULAR MOTOR DISORDER CS=7
DOWNBEAT NYSTAGMUS; WAVEFORM CS=14
NYSTAGMUS CS=2
DOWNBEAT NYSTAGMUS CS=8
CONGENITAL NYSTAGMUS CS=2
NYSTAGMUS CS=3
NYSTAGMUS CS=3

Fig. 8

Fragment of the 1983 bibliographic coupling network. The boxes contain titles of publications with authors' names; title phrases containing words in common with titles in adjacent boxes are displayed in upper case and the common words are shown also on the lines representing coupling links. Coupling strength is indicated for each link.

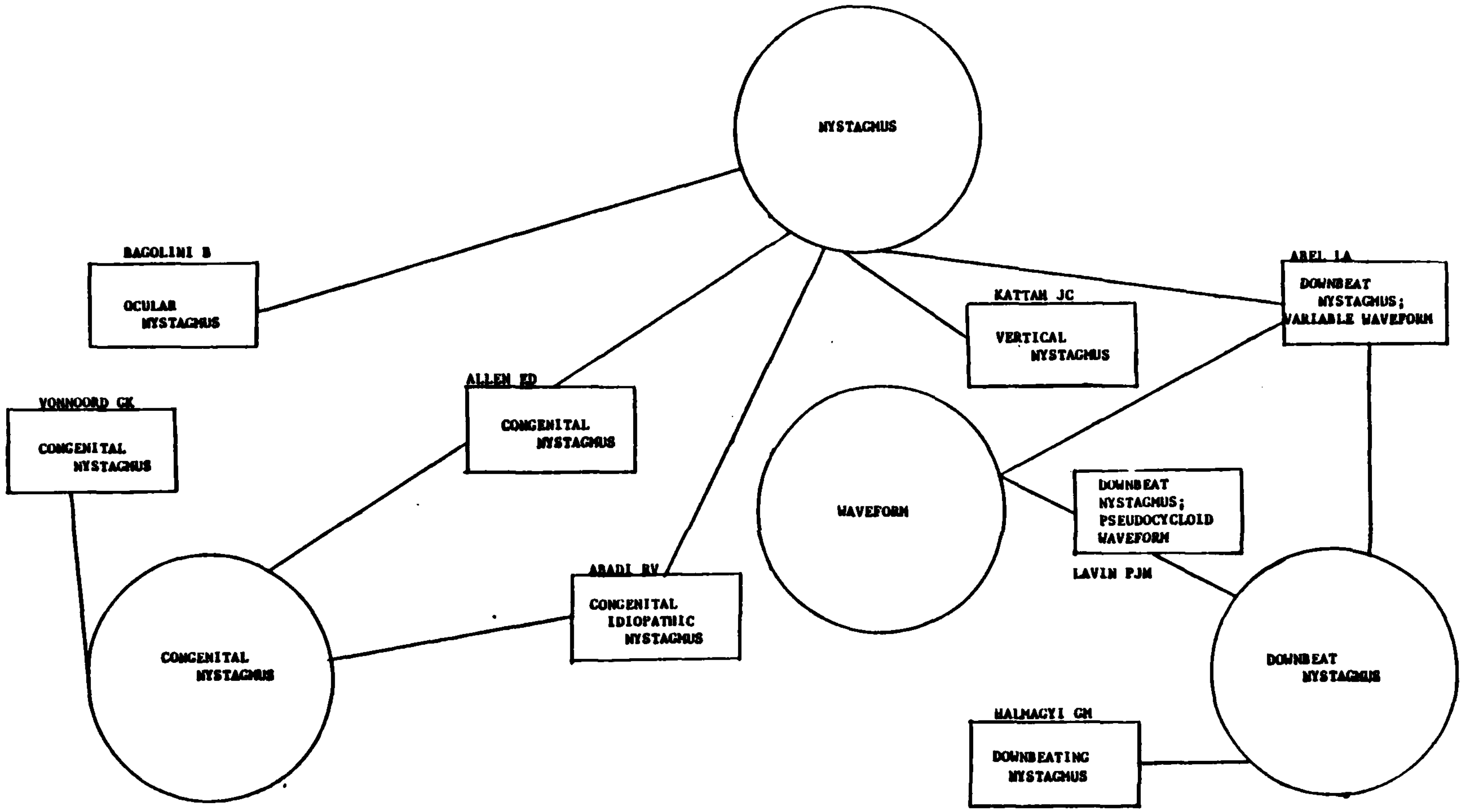


Fig. 9

Bibliographic coupling network fragment with nodes corresponding to shared concepts. The boxes correspond to titles (fig.8) from which only the phrases containing shared words are displayed. Each shared word (shown on fig.8 on coupling links) generates a node (shown as a circle) linked to all titles in which it appears. The links between formerly adjacent boxes are replaced by pathes via shared concept nodes.

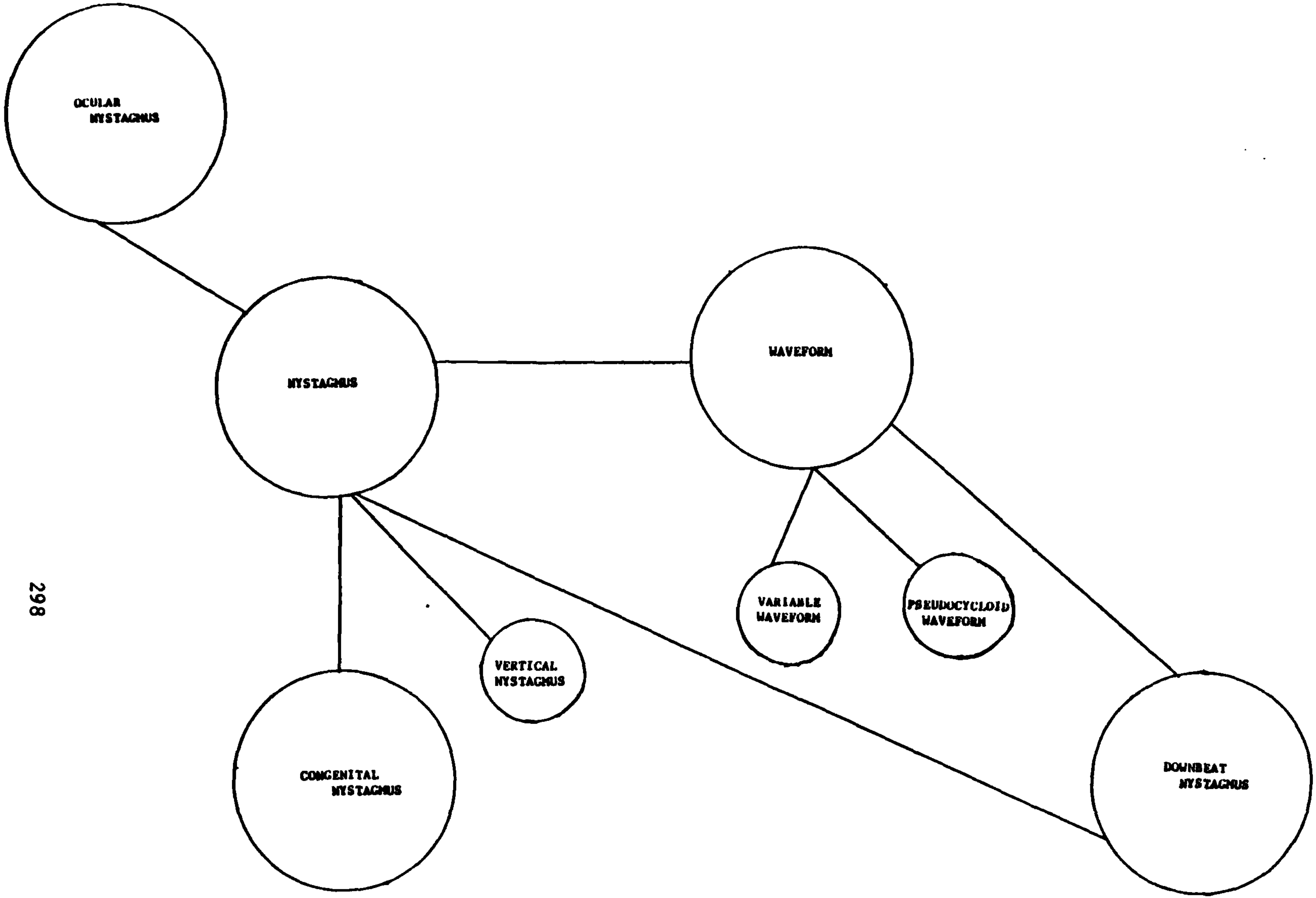


Fig. 10

Concept dependency diagram fragment generated from a biblio-coupling network fragment. Pathes leading on fig. 9 from one concept node to another concept node via boxes have been replaced by a single link between the concept nodes and the boxes eliminated. The boxes linked to only one concept node have been replaced by concept nodes (smaller circles) corresponding to phrases displayed in them on fig. 9.

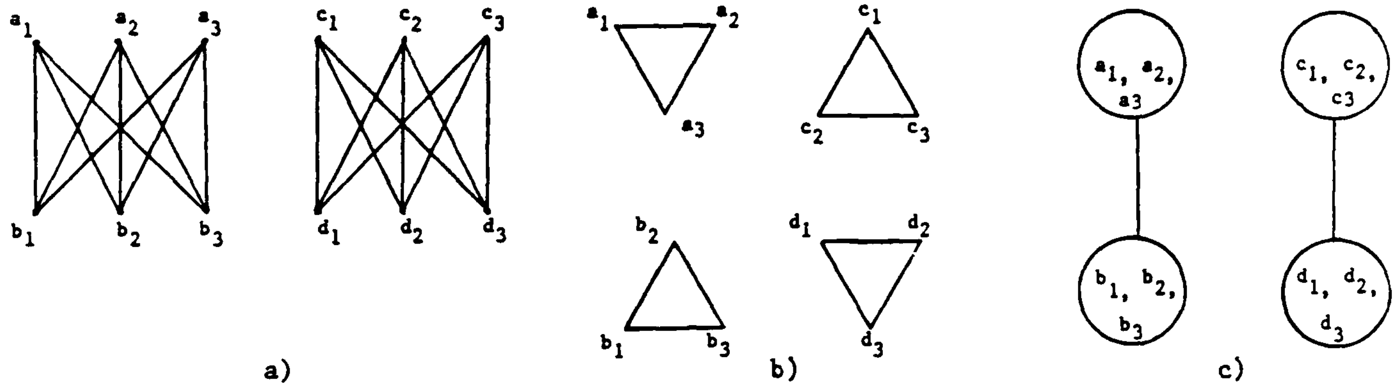


Fig. 11

a) The word co-occurrence graph for texts reporting facts A-B and C-D, where a_i, b_i, c_i and d_i are correspondingly synonymic words expressing A, B, C and D.

b) The graph corresponding to pathes of length 2 (second order associations) in graph a).

c) The graph of the linkages between connected components of b).

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered) AD A 158 285

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
---

2. GOVT ACCESSION NO.
AD-A

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
Data Base Management: Proceedings Of A
Conference, November 1-2, 1984

5. TYPE OF REPORT & PERIOD COVERED
TECHNICAL REPORT

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
Herbert Solomon and Elliot Weinberg
Co-Editors

8. CONTRACT OR GRANT NUMBER(s)
N00014-76-C-0475

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Department of Statistics
Stanford University
Stanford, CA 94305

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
NR-042-267

11. CONTROLLING OFFICE NAME AND ADDRESS
Office of Naval Research
Statistics & Probability Program Code 411SP

12. REPORT DATE
July 31, 1985

13. NUMBER OF PAGES
300

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)

15. SECURITY CLASS. (of this report)
UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Database Management, Artificial Intelligence, Bibliometrics

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A series of eight papers on various aspects of database management. Topics include databases, artificial intelligence, and bibliometrics prepared by experts in these subjects.